Advances in Intelligent Systems and Computing 286

Wojciech Zamojski · Jacek Mazurkiewicz
Jarosław Sugier · Tomasz Walkowiak
Janusz Kacprzyk *Editors*

Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland

Springer

Advances in Intelligent Systems and Computing

Volume 286

Series editor

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

For further volumes:
http://www.springer.com/series/11156

About this Series

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

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

Advisory Board

Wojciech Zamojski · Jacek Mazurkiewicz
Jarosław Sugier · Tomasz Walkowiak
Janusz Kacprzyk
Editors

Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland

Springer

Editors
Wojciech Zamojski
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Wrocław
Poland

Jacek Mazurkiewicz
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Wrocław
Poland

Jarosław Sugier
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Wrocław
Poland

Tomasz Walkowiak
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Wrocław
Poland

Janusz Kacprzyk
Polish Academy of Sciences
Systems Research Institute
Warsaw
Poland

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-3-319-07012-4 ISBN 978-3-319-07013-1 (eBook)
DOI 10.1007/978-3-319-07013-1
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014939038

Preface

We are pleased to present the proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, which took place in a beautiful Brunów Palace, Poland, from 30^{th} June to 4^{th} July, 2014.

Started in 2006, DepCoS – RELCOMEX is a conference organized annually by the Institute of Computer Engineering, Control and Robotics (CECR) from Wrocław University of Technology. Its roots go nearly 40 years back to the heritage of the other two cycles of events: RELCOMEX (1977 – 89) and Microcomputer Schools (1985 – 95) which were organized by the Institute of Engineering Cybernetics (the previous name of CECR) under the leadership of prof. Wojciech Zamojski, now also the DepCoS chairman. In this volume of "Advances in Intelligent and Soft Computing" we would like to present results of research on selected problems of complex systems and their dependability. Effects of the previous DepCoS events were published in volumes 97, 170 and 224 of this series.

Today's complex systems are integrated unities of technical, information, organization, software and human (users, administrators and management) resources. Complexity of such systems comes not only from their involved technical and organizational structures built on hardware and software resources but mainly from complexity of information processes (processing, monitoring, management, etc.) realized in their specific environment. In operation of such wide-ranging and diverse systems their resources are dynamically allocated to ongoing tasks and the rhythm of system events (incoming and/or ongoing tasks, decisions of a management subsystem, system faults, "defense" system reactions, etc.) may be considered as deterministic or/and probabilistic event stream. Security and confidentiality of information processing introduce further complications into the modelling and evaluation methods. Diversity of the processes being realized, their concurrency and their reliance on in-system intelligence often significantly impedes construction of strict mathematical models and calls for application of intelligent and soft computing methods.

Dependability is the modern approach to reliability problems of contemporary complex systems. It is worth to underline the difference between the two terms: system dependability and system reliability. Dependability of systems, especially

computer systems and networks, is based on multi-disciplinary approach to theory, technology, and maintenance of the systems working in a real (and very often unfriendly) environment. Dependability concentrates on efficient realization of tasks, services and jobs by a system considered as a unity of technical, information and human assets, while "classical" reliability is more restrained to analysis of technical system resources (components and structures built from them).

Presenting our conference proceedings to the broader audience we would like to express the sincerest thanks to all the authors who have chosen to describe their research here. It is our hope that the communicated results will help in further developments in complex systems design and analysis aimed at improving their dependability. We believe that the selected contributions will be interesting to all scientists, researchers, practitioners and students who work in these fields of science.

Concluding this brief introduction we must emphasize the role of all reviewers who took part in the evaluation process and whose contribution helped to refine the contents of this volume. Our thanks go to, in alphabetic order, Salem Abdel-Badeeh, Andrzej Białas, Frank Coolen, Manuel Gil Perez, Zbigniew Huzar, Jacek Jarnicki, Vyacheslav Kharchenko, Mieczysław M. Kokar, Alexey Lastovetsky, Marek Litwin, Jan Magott, István Majzik, Jacek Mazurkiewicz, Katarzyna M. Nowak, Yiannis Papadopoulos, Oksana Pomorova, Krzysztof Sacha, Ruslan Smeliansky, Janusz Sosnowski, Jarosław Sugier, Victor Toporkov, Carsten Trinitis, Tomasz Walkowiak, Max Walter, Bernd E. Wolfinger, Marina Yashina, Irina Yatskiv, Wojciech Zamojski, and Włodzimierz Zuberek.

The Editors

Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX

organized by

Institute of Computer Engineering, Control and Robotics,
Wrocław University of Technology
under the auspices of prof. Tadeusz Więckowski, Rector

Brunów Palace, Poland, June 30 – July 4, 2014

Programme Committee

Wojciech Zamojski (Chairman)	Wrocław University of Technology, Poland
Salem Abdel-Badeeh	Ain Shams University Abbasia, Cairo, Egypt
Ali Al-Dahoud	Al-Zaytoonah University, Amman, Jordan
George Anders	University of Toronto, Canada
Artem Adzhemov	Technical University of Communications and Informatics, Moscow, Russia
Włodzimierz M. Barański	Wrocław University of Technology, Poland
Andrzej Białas	Institute of Innovative Technologies EMAG, Katowice, Poland
Dariusz Caban	Wrocław University of Technology, Poland
Krzysztof Cios	Virginia Commonwealth University, Richmond, USA
Frank Coolen	Durham University, UK
Antonio Ferrari	University of Aveiro, Portugal
Francesco Flammini	University of Naples "Federico II", Napoli, Italy
Manuel Gill Perez	University of Murcia, Spain
Janusz Górski	Gdansk University of Technology, Poland
Zbigniew Huzar	Wrocław University of Technology, Poland
Igor Kabashkin	Transport and Telecommunication Institute, Riga, Latvia
Janusz Kacprzyk	Polish Academy of Sciences, Warsaw, Poland
Andrzej Kasprzak	Wrocław University of Technology, Poland
Vyacheslav S. Kharchenko	National Aerospace University "KhAI", Kharkov, Ukraine
Mieczysław M. Kokar	Northeastern University, Boston, USA
Krzysztof Kołowrocki	Gdynia Maritime University, Poland
Leszek Kotulski	AGH University of Science and Technology, Krakow, Poland
Henryk Krawczyk	Gdansk University of Technology, Poland

Organizing Committee

Contents

Erratum

Framework for the Distributed Computing of the Application Components

Razvan-Mihai Aciu and Horia Ciocarlie

Department of Computer and Software Engineering, "Politehnica" University of Timisoara
Blvd Vasile Parvan, Nr. 2, Postcode 300223, Timisoara, Romania
razvanaciu@yahoo.com, horia@cs.upt.ro

Abstract. Writing real world distributed applications is a challenging task. Even if well known models or powerful frameworks such as MapReduce or HADOOP are employed, the complexity of the aspects involved, such as specific programming and data models, deployment scripts or a hard debugging process are enough to require many working hours or even make the entire process unsuitable for practical purposes. For applications which need some of their own components to be computed in a distributed manner, a generic model incurs an unnecessary overhead and makes the whole development slower. We propose a MapReduce framework which automatically handles all the distributed computing tasks such as computing resources abstraction, code deployment, objects serialization, remote invocations and synchronizations with only a minimal coding overhead. With only minimal constructions dependable distributed components can be developed and run on heterogeneous platforms and networks. The presented results confirm the performance of the proposed method.

Keywords: distributed computing, serialization, synchronization, invocation.

1 Introduction

Today resource intensive applications can run on networks starting from a few computers and reaching thousands of dedicated servers organized in clouds or grids. The internet also brings the possibility to run applications on global scale networks, many of them forming resources, known as Volunteer Computing Networks [1]. There are many processing resources, which can be used by an application. Network computers, microprocessor cores and GPUs can be used to compute laborious tasks in parallel [2]. However, each of these resources require special handling such as writing threads for CPU cores, kernels for GPUs and serialization/invocation protocols for network resources. From the application level, each resource executes some code to process inputs and to generate outputs. It should be possible to design a construction, which abstracts computing resources in a generic way. In Java such construction can be a common adapter interface between the application logic and the specific resources. The Java basics to run threads on microprocessor cores are the class Thread and the interface Runnable [3]. There are no standard equivalents of these for network

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_1,

computing resources or for GPU cores [4], which imposes challenging tasks for the programmers who want to make use of all available resources, especially if the applications are to be run on ad-hoc, loosely coupled, heterogeneous networks [5].

The current frameworks employ for distributed computing auxiliary components like application servers, Interface Description Language (IDL) compilers, libraries for serialization, network resources discovery, communication protocols and deployment [6]. If we want to execute in a distributed manner application components, the role of some of the above components can be replaced by standard functionality such as Java serialization and generics. Another approach is to use programming languages with strong distributed programming capabilities, such as Erlang [7] but this poses interfacing problems when the cooperation with modules written in mainstream languages is needed. With our Java framework we try to implement a representative package of a HADOOP cluster [8] functionality, when the distributed computing is needed only for application own components, tailoring it for a simplified and transparent usage of local and remote computing resources, using small to medium heterogeneous networks.

As the distributed applications are generally the ones which require a lot of processing power [9], we want to be sure that every resource is fully used. This is why in our tests we use *scalability* and *resources load balancing* as fundamental metrics to evaluate our results [10]. The *ease of use* and *the available features* are equally important for the adoption of our algorithm and framework in production.

2 Algorithm Overview

Our algorithm and framework handles CPU cores and network resources in a generic way and it can also be extended to GPU cores. All the required low level tasks are performed automatically by the framework and the programmer needs only to concentrate on the application logic.

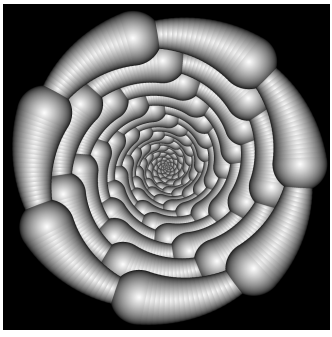


Fig. 1. An image with 3500 spheres rendered with our test program

If we have to render a high resolution complex 3D image as in Fig. 1, using an arbitrary sized computer network, we can divide the image in sufficiently large parts to justify parallel processing when compared with the added network overhead and send

these parts to the available computers when they are idle. This is the best approach for heterogeneous networks or for when the computers number can change in time.

The workload must be divided in pieces, which will be process on the available resources and the partial results will be received and assembled. This follows the well known MapReduce model [11], on which the Map phase is represented by the distributed computations and the Reduce phase is represented by the assembly of the received results back on the application. The algorithm in pseudocode is represented in Fig. 2:

```
(1)     var sd:Scheduler
(2)     sd=new Scheduler(computeClass,initData,destination)
(3)     for piece=every piece of the workload
(4)             sd.addInvocation(destinationPosition,piece)
(5)     sd.waitForAll()
(6)     assemble_received_results()
```

Fig. 2. Algorithm pseudocode

An invocation is a task scheduled for execution on an abstracted computing resource, local or remote. The *Scheduler* is a framework component, responsible for the handling of all low level details involved on the distributed computing. On its initialization, the *Scheduler* needs a class which implements the required computation (*computeClass*), a constant global data which will be passed at the initialization of all distributed computations (*initData*) and a holder for results (*destination*).

The method *addInvocation* executes asynchronously and it enqueues an invocation to the invocations list. Every invocation has the data needed to perform its computation (*piece*) and an abstract place in destination (*destinationPosition*) where the results are returned. When invocations are enqueued, the Scheduler starts to create worker threads, each one connected to a computing resource.

The framework provides a special server to which the workers connect. Every computer which takes part in computation must have a running server on it. The worker threads get invocations from queue, run them on their computing resource and return the results. For this, the Scheduler must first deploy the necessary code (*computeClass* and all its dependencies) to the remote computers and also *initData*. These are sent only once, no matter how many invocations will run on that computer. The code deployment is done by using a specialized class loader and for data a serialization engine must be employed. In our implementation we use the standard Java serialization framework.

After all the workload parts are scheduled, the method *waitForAll* is used to wait until all the computations are performed and the results are retrieved. When the results are received, they will be assembled according to the application logic. These are all the necessary steps. The details needed in a traditional distributed application, like network discovery, code deployment, serialization and synchronizations are abstracted from the application logic. Moreover, the computing resources are also abstracted, so the application can automatically make use of any resource, such as local microprocessor cores or network computers.

In our example, every image line is an invocation, so for a 2000 lines image we have 2000 invocations. The initial data is the scene itself (including output resolution,

observer position, view angles), because these are invariants. There is no defined order for invocations processing, so their results need to be stored in order to be assembled in the right order in the final image. The *computeClass* is a class derived from a special interface (provided by framework), which handles the actual image line computation. This class and all its dependencies will be deployed to network and it will be run in a distributed manner. The *destinationPosition* is the index in the vector of image lines where the invocation result will be put. The *piece* is an invocation specific data, in our example the vertical angle for each image line.

3 Detailed Description for Framework

The in-depth description gives for each step involved in the algorithm the full extent of the options and customizations that can be made, the required framework support, the suggestions regarding the implementation on different platforms.

A. Network setup

Every computer, which is part in the distributed computing process, must run a specialized server. This server allows queries regarding its version and available computing resources. A server allows a maximum number of concurrent connections at most equal with its number of computing resources. The scheduler creates a worker thread only if there is a server with available connections and once the connection is established between the worker and the server, it remains open until all jobs end or until an exception occurs. In this way a computer core is assigned to a single worker in order to fully use all the resources and in the same time to minimize kernel threads switching. This model works well for small to medium networks, with a top of simultaneous open sockets of around some thousands. It is also preferred when the application is on a private network, which cannot be directly accessed from outside.

For larger networks an alternate model can be used, based on regular queries (pings) to the servers to which computations were sent, in order to check the status and retrieve results. In this case there is no bound in regard to the maximum open sockets number, because a socket would exist only during a query.

To address unsecure, regulated or volunteer networks, additional requirements must be observed: application & server authentication, communication encryption, possibility to set upper bounds on the usage of the server resources and security policies to access file system, network or other security or privacy sensitive functions.

B. The distributed executable code

The application components, which are designed to run in a distributed manner (*computeClass* from Fig. 2) must implement the following interface:

```
(1)   public interface Distributed
(2)           <InitData,Index,RunData,RetType>{
(3)   boolean    dInit(final InitData initData);
(4)   RetType    dRun(final Index idx,RunData runData);
(5)   }
```

Fig. 3. The Distributed interface

Through the framework, Java generics are employed to ensure type safety. Every server connection creates its own instance of *computeClass* so at maximum on a server can be as much *computeClass* instances as its total cores number.

Method *dInit* is called only once, when the new instance is created. It returns true if the initialization was successful and the worker can use the new instance. The argument *initData* is the same for all invocations and must invariable. The newly created instance will be used for all computations from that worker. In this way it becomes possible for *computeClass* to keep state information between invocations, such as for caching partial computations. The order or the number of invocations handled by this particular instance is not specified.

Method *dRun* is called for every invocation. The argument *idx* is an abstract index in the destination. It can be anything: an index into a vector or a key into a map. It keeps track of the results order. It must be unique for every invocation. Our test application uses the image lines indexes as *idx*. The argument *runData* is used to pass specific arguments for each invocation. On success, *dRun* returns a newly created object with the computation result. If *dRun* returns null, an error is signaled.

The classes which implement *Distributed* cannot have a common memory area (like static variables), because they can run on different hosts. This requirement can be enforced at runtime by analyzing the used members and their dependencies.

C. The scheduler instantiation

The *Scheduler* class is provided by the framework. It has the following signature:

public class Scheduler<InitData,Index,RunData,RetType>

The generic parameters *InitData*, *Index*, *RunData* and *RetType* were described in section III.A. *Scheduler* has both a static and a non-static constructor. The static constructor is used for automatic system wide initializations, such as network discovery. This check is made once, at the application start. Taking into account the dynamic nature of the network, which allows computers to be added or removed any time, subsequent resource checks are also possible, started by the programmer or automatically performed at specific intervals of time. The non-static constructor for *Scheduler* has the following signature:

public Scheduler(final Class<?> distributedClass,
final InitData initData, Destination<Index,RetType> dst)

The argument *distributedClass* is the class representation of the distributed class. Its code and all its dependencies are sent to the available servers to be run remotely. This class implements the interface *Distributed,* so it can be called in a standard way. In languages with full reflection such as Java or C#, the serialization of a class description and its methods code can be achieved using the provided standard API.

The argument *initData* is the initial constant data, to be used at the initialization of each distributed workers. It is sent to every server only once.

The argument *dst* is the destination of the distributed computations results. The interface *Destination* is detailed in Fig. 4 and it defines an abstract destination.

```
(1)     public interface Destination<Index,RetType>{
(2)             void    set(Index idx,RetType ret);
(3)     }
```

Fig. 4. The Destination interface

When a call to *dRun* finishes, the result is sent back and the method *set* of *dst* is called with the destination index and the computation result. The destination class can have different behaviors, according to the application logic. If all the computations results must be retrieved first (as in our example) then the destination can be a thin wrapper over an array or collection class. If the computations results can be processed separately, the call to *set* can directly encapsulate the processing of each result.

D. The invocations and the workers

An invocation is a computation scheduled for execution. It is added to the invocations list with the *Scheduler* method *addInvocation*:

public void addInvocation(Index idx,RunData runData)

The argument *idx* is the result index in *dst*. The argument runData is the specific data necessary for this computation. The method *addInvocation* runs asynchronously, so it does not block the application loop. It puts the invocation in an invocations list.

If all the existent workers are busy and there are more available computing resources, *addInvocation* creates a new worker to process the newly added invocation. A worker is a thread created by *Scheduler* which handles all the communications with a specific resource. A worker does not compute invocations, but it only sends them to the associated resource, receive their results and put them in destination. As such, a worker thread consumes very few resources and there can be thousands of workers.

First, on its creation, a worker connects to a resource and locks it for itself. After that an instance of *distributedClass* is created remotely and the *initData* is passed to its *dInit* method. This instance will be kept alive during the worker's life. After that, the worker takes invocations from list and sends their data (*idx* and *runData*) to the associated resource. There, the data is passed to the *dRun* method of the *distributedClass* instance, it is computed and the result is returned.

The errors caused by the application logic itself are signaled back with exceptions. On errors caused by the network the worker first tries to reconnect and if it fails it informs the framework to check if the remote server is still available. If the remote server cannot be discovered, it is removed from the available servers list. If the worker cannot reconnect, it will shut down itself. In this case the current invocation will be put back to the invocations list, in order to be processed by another worker.

E. The end of the distributed computations

After all the invocations were scheduled, the application has two choices to wait for their completion. The easier choice is to call the *Scheduler* method *waitForAll*:

public void waitForAll()

This is a blocking method, which waits for the completion of all invocations, including the ones from the invocations list and the ones currently running. When *waitForAll* returns, it is guaranteed that all the invocations were computed and the

results were passed to destination. The other choice is to manually check for completion, using the following methods of *Scheduler*:

public int getAddedInvocationsNb()

public int getCompletedInvocationsNb()

The method *getAddedInvocationsNb* returns the total number of invocations added to the scheduler using *addInvocation*. The method *getCompletedInvocationsNb* returns the number of the invocations, which were already successfully computed. By using these two methods, the application knows the scheduled invocations status.

When all the invocations are computed, the scheduler stops all worker threads. A worker signals to the associated server to free the remote resources and ends.

4 Theoretical Model

The model analyzes the theoretical execution time which can be achieved with our algorithm, both on local cores and on network. Different factors are taken into consideration, such as remote connections startup time and network speed. In the case of heterogeneous networks, different hardware configurations lead to different computation times. More than that, not all the invocations require the same amount of time. In our example, an image line with more spheres intersections is rendered slower than one with fewer spheres.

We define the *invocation average computation time* (T_A) metric as the *total computation time* (T_T) averaged to the *total number of invocations* (I_T). We consider a hardware reference (H_R) and T_T is a function of if it. $T_A = T_T(H_R)/I_T$. T_T is considered for only one core on a specific configuration. In that case: $T_T(H_R) = T_A * I_T$.

For any computer, we define *relative speedup to the hardware reference* (S), where S is a function of the other hardware: $S(H_i) = T_T(H_i)/T_T(H_R)$. When the application is run simultaneously on multiple independent cores, the total computation time is the maximum time of the partial computations: $T_T = \max(T_{Ti})$. We have:

$$T_T = \max(T_A * S_i * I_i) \quad (1)$$

Where S_i is $S(H_i)$ and I_i is the number of invocations run on that specific core. If the application runs on network, for every invocation we define a *remoting overhead time* (T_O) given by the data serialization and the network speed when the parameters are sent and when the results are received. There is also a *connection setup time* (T_C) necessary to establish the socket connection, to run the server handler for the connection and to dispose all these when the worker ends. In that case (1) becomes:

$$T_T = \max((T_A * S_i + T_O) * I_i + T_C) \quad (2)$$

On the optimal case, if we have I_T cores so every invocation will run on its own core, (2) becomes:

$$T_T = \max(T_A * S_i + T_O + T_C) \quad (3)$$

As $T_O + T_C$ is constant, from (1) and (3) it can be seen that the optimal case is when every core runs only one invocation, so the $T_O * I_i$ becomes T_O. In that case, every available core is used and in the same time the network traffic and the associated serialization overhead is reduced to minimum. This can be achieved by having a

number of invocations equal with the available cores (the optimum case), or by having very intensive invocations, so most of the time will be spent in computations (the T_A*S_i term), with only a short percentage of time spent in network related tasks.

The above conclusions are true when the resources are reliable (the invocations succeed). Else, it is more advantageous to have lighter invocations (as computing time), so their re-computation would be cheaper.

5 Practical Results

The algorithm and framework were tested both in a computer network and on a computer with a quad core microprocessor. Two important metrics were especially evaluated. First is the *total speedup when new resources are added*. This metric also gives a good evaluation if it is advantageous for a certain application to use more resources, taking into account other factors like their economic costs. The other metric is the *workload distribution on each computing resource* – this is important to evaluate the ability of the algorithm to distribute the workload on the existing resources, especially in the case of heterogeneous networks. In our tests we also tried different operating systems and Java implementations in order to assess how they are working with our framework. For all tests we used an application which renders the image from Fig. 1 with a resolution of 2000x2000 pixels. An invocation is made from an image line, so we had 2000 invocations. At every test a fresh server was run in order to clear the cached remote classes to have similar startup conditions.

A. Tests on a computer network

We used a wired network of 10+1 computers, with Intel® Core™ 2 6600@2.40GHz CPU, running Kubuntu 8.04 on 64 bits, with Java HotSpot Server VM 1.6.0_06. One computer was used only for the base application and the invocations were allowed to run only on the other computers, in order to obtain homogenous results from all involved resources. We started with one computer and on each iteration we added new computers, measuring the relative speedup to the case of only one. As each microprocessor has two cores, finally we had 20 cores to run our invocations. The speedup is shown in Fig. 5 and the workload on every core is shown in Fig. 6.

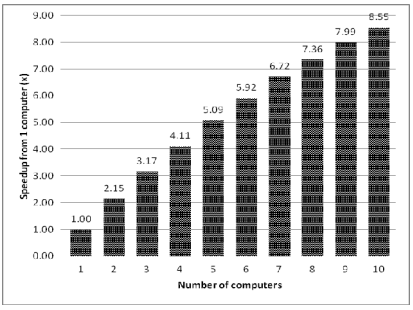


Fig. 5. Speedup on network

From Fig. 5 it can be seen that on a small number of computers, the speedup is close to the optimum. The experimental results for 2-5 computers show a performance slightly over the theoretical model due to external factors which influence the measurements of small time intervals, like the variance of the network traffic.

When the computers number grows, the speedup is lower because the computations finished very quickly (around 1s) and other factors like the network discovery and sockets and threads management (the T_C component from the theoretical model) counted for a bigger part from the total run time.

The workload distribution was close to the optimum (the case with equal amount of invocations run on every core). The average percent difference on all cores was maximum 1.03% for all test runs and the maximal percent difference of a core workload from the optimum was 2.8%.

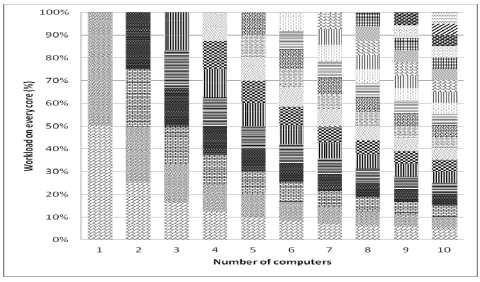


Fig. 6. Workload on each core on network

Tests on a Computer Cores

We used a computer with Intel® Core™ 2 QUAD Q6600@2.40GHz CPU, running Windows Vista Business SP2 on 32 bits, with Java HotSpot Client VM 1.7.0_11-b21. This computer has four cores. We started by allowing the invocations to run on only one core and on each iteration we added new cores. The speedup is shown in Fig. 7 and the workload on every core is shown in Fig. 8.

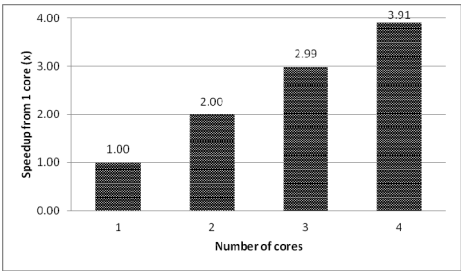


Fig. 7. Speedup on a computer

It can be seen from Fig. 7 that on running on local cores the speedup is very close to the optimum. Only on the 4th core there is a somewhat larger (0.09%) difference to the optimum. This is due to the fact that this core also needs to run the main application

(the scheduler and all the synchronization code). This result shows that the scheduler and all its associated threads (the workers) are consuming very few resources, as they mainly only send the invocations parameters and wait for results.

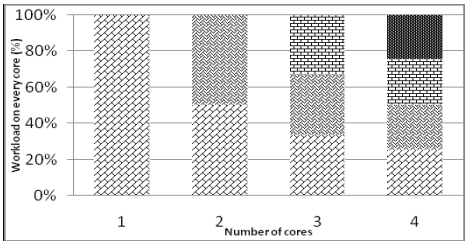


Fig. 8. Workload on each core on a computer

The workload distribution between the different cores had a maximum percent difference from the optimum of 0.6% (due to the different load on the last thread) and the average percent difference on all test runs was maximum 0.4%.

6 Conclusion

The proposed algorithm and framework makes possible to use automatically deployed application classes as distributed components. The framework abstracts the resources such as local CPU cores and computers from heterogeneous networks and it allows the programmer to use them transparently, in a uniform manner. The framework is suitable for many types of applications. It works well for languages which run on virtual machines, such as Java or C#, but with some restrictions it can be used for languages compiled to native code and without advanced reflection, such as C/C++.

The framework usage is very simple. In the first step the programmer needs to implement the *Distributed* interface on the class he wants to execute in a distributed manner. In the second step, the programmer adds invocations to a scheduler. From this point, all the distributed tasks such as network management, serialization, deployment and synchronization are automatically performed.

We provided a Java implementation for framework. From the practical results it can be seen the algorithm is scalable, both in term of local cores and network computers and it provides a good load-balancing, by uniformly distributing the tasks to all the available resources. In all tests the framework was proved to be dependable and the final result was provided even on the occurrence of network errors.

Our algorithm and framework open many research directions and we consider developing them further to use GPU cores, to improve the overall reliability on errors and to interoperate with other distributed computing frameworks.

References

1. Fedak, G.: Desktop Grid Computing. Chapman and Hall/CRC (2012), doi:10.1201/b12206-16
2. Feinbube, F.: Programming models for parallel heterogeneous computing. In: Proceedings of the 5th Ph.D. Retreat of the HPI Research School on Service-Oriented Systems Engineering (2011)
3. Lea, D.: Concurrent Programming in Java: Design Principles and Patterns. Addison-Wesley Professional (2003) ISBN-10: 0-201-31009-0
4. Espeland, H., Beskow, P.B., Stensland, H.K., Olsen, P.N., Kristoffersen, S., Griwodz, C., Halvorsen, P.: P2G: A framework for distributed real-time processing of multimedia data. In: 40th International Conference on Parallel Processing Workshops, ICPPW (2011)
5. Krasic, C., Saubhasik, M., Goel, A., Sinha, A.: Fair and timely scheduling via cooperative polling. In: EuroSys (2009)
6. Pop, F., Grigoras, M.V., Dobre, C., Achim, O., Cristea, V.: Load-balancing metric for service dependability in large scale distributed environments. Scalable Computing: Practice and Experience 12(4), 391–401 (2011)
7. Armstrong, J.: Programming Erlang: Software for a Concurrent World, 1st edn. (2007)
8. Kiran, M., Kumar, A., Mukherjee, S., Ravi Prakash, G.: Verification and Validation of MapReduce Program Model for Parallel Support Vector Machine Algorithm on Hadoop Cluster. IJCSI International Journal of Computer Science Issues 10(3(1)) (2013)
9. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems 28 (2012)
10. Pandey, S., Buyya, R.: Scheduling workflow applications based on multi-source parallel data retrieval in distributed computing networks. The Computer Journal (2012)
11. Lämmel, R.: Google's MapReduce programming model – Revisited. Science of Computer Programming 70(1) (2008)

Analysis of Statistical Characteristics of User Arrival Process to the Testing Service

Artem Adzhemov, Nikolay Albov, and Irina Sineva

Moscow Technical University of Communication and Informatics,
8a Aviamotornaya St., Moscow, Russia, 111024
{asa,iss}@mtuci.ru, nick.albov@gmail.com

Abstract. As a part of e-Learning system the testing service allows to measure students' skills. In order to design testing service it is necessary to study interaction between a student as the user of service and the testing service. We measure and analyze the time between requests and the testing service. By using KS-method we fit the measurement results to the theoretical probability distribution. The conducted analysis shows that the use of a normal model for analyzed data is not suitable. It is found that the inter-request time distribution is the log-logistic distribution. The estimates of the distribution parameters will be suitable for all such interactions between the user and the testing service. The homogeneity hypothesis for inter-request times is verified. The nonparametric Kruskal-Wallis test is applied as a homogeneity test. Then in order to aggregate observations into larger groups the cluster analysis of the log-logistic distribution parameters is carried out. The following methods of the cluster analysis are used: an agglomerative hierarchical clustering method and a k-means method. The results of our study can be used for the modeling of computer-based testing service.

Keywords: inter-request times, think-times, distribution fitting, log-logistic distribution, KS-method, Kruskal-Wallis test, cluster analysis, e-Learning, testing service.

1 Introduction

Information and communication technologies (ICTs) have essentially expanded the list of services offered to the people. Today the new technologies are closely connected with the computer and are widely used. Such as: e-Learning, e-Medicine, e-Government, e-Commerce etc.

Recent developments in ICTs, such as the Internet and World Wide Web, enable increased production and dissemination of information across geographical boundaries. We are witnesses of formation of the new world – the world in which the person widely uses computer in the everyday activity. The virtual world created by the people should correspond to the real-life world. Each subject in the virtual world should have a prototype in the real world. For example, the dialogue in the real world corresponds to the chat in the virtual-world.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_2,

Moreover, additional ICTs present the new possibility that in the real world does not exist. Such as: hyperlinks, personal time-table, selecting the teacher in the real-time etc.

Now it is easy to find different types of educational portals in the Internet that introduce different types of services that cover all area of human life. Some of them introduce only the access to the electronic document, others present interactive educational service. These services are different only in the process of interaction between the user and the system. These types of services have various graphic design and different algorithm of interaction with users. Depending on a required document users can access to services in different ways. That results in different workload on the server resource.

The given paper deals with the investigation of test or quiz systems which are widely used in e-learning system. There are different types of computer-based testing system. In this paper we analyze partially adaptive computer-based testing system. For optimal design of computer-based testing system it is necessary to do researches of workload on such system. We characterize user behavior in terms of inter-request time or think-time. As the information transfer time over network is much less than the user think-time we consider that the time between a request and user think time is approximately equal. Although arrival process can be described in more details, the inter-request time is the main characteristic of such process. The statistical analysis of aggregated arrival process to the testing service was previously described by the authors in [8]. The statistics obtained in our research can be used in the future as parameters for analytic and simulation models describing such traffic. Also the results of analysis can be used to evaluate the performance of test service, improve the network management and plan test service capacity. The time between two consequent requests to user test system depends on user subjective qualities: knowledge level, his mental and physical condition at the time of testing, the complexity of the test.

The system under consideration consists of an assessment engine and an item bank. The assessment engine includes software and hardware which are necessary to create a test. An item bank stores tests. The engine uses the bank to generate a test. Test-takers can request and answer an item or skip it and then go to the next item. In this test system there is the following limitation: the time limit is established by the test designer.

2 Background

The following statistical hypotheses were tested during data analysis:

1) The hypothesis of the homogeneity of users inter-request time distribution within the group.
2) The hypothesis of inter-request time distribution for each user test session.

To determine the distribution which is best fitted to experimental data we use the Kolmogorov-Smirnov Goodness-of-Fit Test [1]. The Kolmogorov-Smirnov (K-S) test is useful in deciding if a sample comes from a population with a specific continuous

distribution. We use several distributions in order to check which one would fit the inter-request time better. In this paper, the significance level is considered as the degree of closeness of the analyzed data sets to the theoretical probability distribution. Significance level with value less than 0.05 indicates that the tested data is significantly different from the hypothesized distribution. The obtained high significance level allows using the results of the research as a probabilistic model for the same interaction. We use the maximum likelihood estimation (MLE) method to evaluate the parameters of the distributions. For the examination of homogeneity among grouped data a Kruskal–Wallis test [1, 6] is conducted due to asymmetric inter-request time distribution. The Kruskal–Wallis test is a nonparametric method that compares the medians of two or more samples to determine if the samples have come from different populations. The check for homogeneity provides valuable information about the groups taking the test, individual tests. It should be noted that only the requests to test service were analyzed and the technical implementation of the test system was ignored.

Probability distributions were chosen looking on the type of histograms constructed on empirical data and based on other well-known works of the study of traffic and information about user think time. The student's think time is well modeled with logarithmic type of distributions. Logarithmic dependence is observed in a number of studies, for example, the log-logistic distribution for length of HTTP request and idle time between messages [2]; the log-normal distribution for characterization of some network metrics [3], the log-normal distribution for reading time - the interval between two consecutive Web page requests [4], the lognormal distribution for the test item response time [5].

3 Data Analysis

In this section we discuss data analysis. Let's consider several groups of students getting quiz during one academic year. Three groups of students in three different disciplines such as humanities, technology and natural science were chosen to take tests during the session.

The humanities test will be discussed in more details as an example of study design. Fifteen students took part in this test. The test set consisted of 64 test items. The test duration was 45 minutes, or an average of about 0.7 minutes per one test item. The number of observations for each user was from 64 to 86. We calculated basic statistics data. As a result we got the following statistics: sample mean from 11.8 to 28 seconds, sample median from 9.9 to 20.5 seconds, the value of the asymmetry coefficient from 3.1 to 7.3. According to this statistics it can be assumed that the data is heterogeneous, and furthermore it is proved by the exact calculations.

The study of observations resulted in several patterns of temporal students' behavior. The behavior patterns are listed below: 1) time distribution is grouped around the mean value with a positive coefficient of asymmetry and rare observations exceeding the mean value more than three sigma; 2) time distribution is grouped around the mean value with a positive coefficient of asymmetry observations without exceeding the mean value more than three sigma.

3.1 Student Behavior Patterns

Let us consider the first pattern of students' behavior. As we have noted above that student think time is close to inter-arrival time so in further reasoning we will assume they are the same. The student takes test without skipping test items and spends an average time needed for thinking. This time is close to the average time on all observations for a given student. This think time distribution has a positive skewness due to the fact that on some test items students spend more time. This kind of behavior is observed in two variants: the first variant occurs when one observation exceeds the average for all observations. Such type of observation is usually found at the end of the test session. The second variant can be observed in several places of the test session. Figure 1 shows one data set that contains this type of observation. It is clear that there is only one value greater than 200 seconds. This value is a statistical outlier. However, it should be noted that the exclusion of such observation from the data set does not significantly affect the results of the analysis. Exclusion of these observations from the analyzed data set may result in loss of correctness of the model of users' behavior.

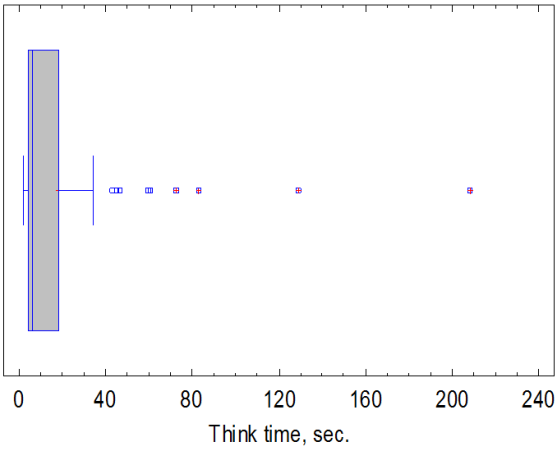


Fig. 1. Box-and-Whisker plot for a single set of observations with outlier

In the second pattern of students' behavior the variability of think time values isn't very large. Such type of users skips fewer test items during the test session. At the time of the analysis a rare observation was found to which we could not find distribution. For example, such type of students whose think time was between the consequently requests had multimodal time distribution. This can be explained by the fact that students spend different time on different test questions. For example, we found one observation where the user answer time lay in the range between twenty and forty seconds and observed small series of intervals when student skipped the test question.

3.2 Statistical Data Analysis

During the analysis data of 173 test session were processed. Initial data include 9299 student requests for test item. Inter-request time distributions for each test session were found. Our choice of distribution is based on previous knowledge and the fact that most of the observations have a positive skewness and the sample mean is greater than the sample median. The inter-request time positive skewness did not allow to use a normal model for describing data. For the selection of the theoretical density distribution functions several different distributions were used: log-logistic, lognormal, Weibull, Gamma, Birnbaum – Saunders, inverse Gaussian distribution, the distribution of extreme values. The K-S test showed that the log-logistic distribution was best fitted for our model parameters based on the data set. Therefore the result of the log-logistic distribution appearance as the best fitted distribution is not obvious and proven on the basis of general considerations. The probability density function for the log-logistic distribution is given by the following formula:

$$f(x)=\frac{1}{\sigma x}\frac{e^{z}}{[1+e^{z}]^{2}},\ \text{where}\ \ z=\frac{\ln(x)-\mu}{\sigma} \tag{1}$$

The distribution (1) depends on two parameters: μ and $\sigma>0$. Expectation and variance in terms of these parameters and gamma function are the follows:

$$E(X)=e^{\mu}\Gamma(1+\sigma)\Gamma(1-\sigma) \tag{2}$$

$$Var(X)=e^{2\mu}[\Gamma(1+2\sigma)\Gamma(1-2\sigma)-\Gamma^{2}(1+\sigma)\Gamma^{2}(1-\sigma)] \tag{3}$$

This distribution is also characterized by two other parameters: the parameter $\alpha=\exp(\mu)$ is a scale parameter and the parameter $\beta=\sigma^{-1}$ is a shape parameter.

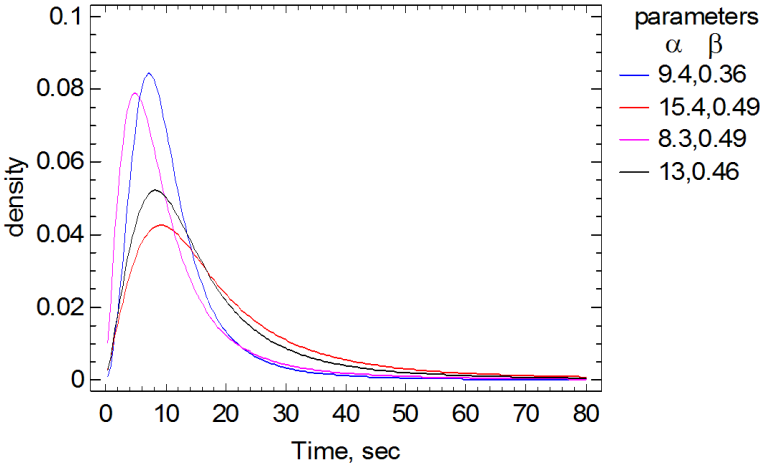


Fig. 2. Distribution density function with different parameters

The density distribution function with different parameters is shown on Figure 2. These distribution parameters were obtained from performed cluster analysis of distribution parameters characterizing the humanities discipline.

The significance levels (p-value) of 0.14 to 0.98 were obtained by testing hypothesis about the log-logistic distribution. The estimates of the parameters for log-logistic distribution were respectively in the range 7.8 ÷19 (α) and 0.3 ÷ 0.57 (β).

3.3 Cluster Analysis

All observations were checked for homogeneity in order to further aggregation into larger groups. Nonparametric Kruskal-Wallis tests were conducted on the observations to verify homogeneity assumptions. Subsequently we found that the observations belonging to the group were not homogeneous. Then we used the clustering analysis [7] to search for homogeneous groups. An agglomerative hierarchical clustering method and a non-hierarchical iterative clustering or so called k-means method were used. For the agglomerative hierarchical clustering we used various methods to create the cluster: nearest neighbors, furthest neighbor, centroid, median, Ward method, group average. The Euclidean squared distance was used for k-means method. The K-means cluster analysis was applied to the following data: sample mean, sample median, sample standard deviation. As a result, the smallest number of subgroups was found using the Ward method. Quadratic Euclidean distance as the distance between objects for Ward method was used. Table 1 shows centroids that were obtained after clustering the data. Figure 3 shows a scatter plot with two clustering parameters: sample mean and standard deviation. It should be noted that cluster structure is not sensitive for options of clustering procedure.

Table 1. Cluster centroids for humanities discipline

Cluster number	Sample mean	Sample median	Sample standard deviation
1	12.3438	8.93687	10.2152
2	25.1784	15.5503	36.1014
3	15.6277	7.30692	25.7327
4	18.5948	12.5352	20.3671

For each subgroup homogeneity of Kruskal-Wallis test and K-S test for checking compliance with the log-logistic distribution were conducted. While checking the homogeneity of the subgroups we found out four subgroups with p-values 0.2573, 0.0952, 0.3485 and 0.6493 respectively. K-S test provides log-logistic distributions as the most appropriate for different clusters. More precisely results looks as follows: the first subgroup - p-value is 0.688982 , distribution parameters - $\alpha = 9.43511$, $\beta = 0.361293$, the second subgroup - p-value is 0.561375 , distribution parameters - $\alpha = 15.3688$, $\beta = 0.49033$, and the third subgroup - p- value is 0.0741992,

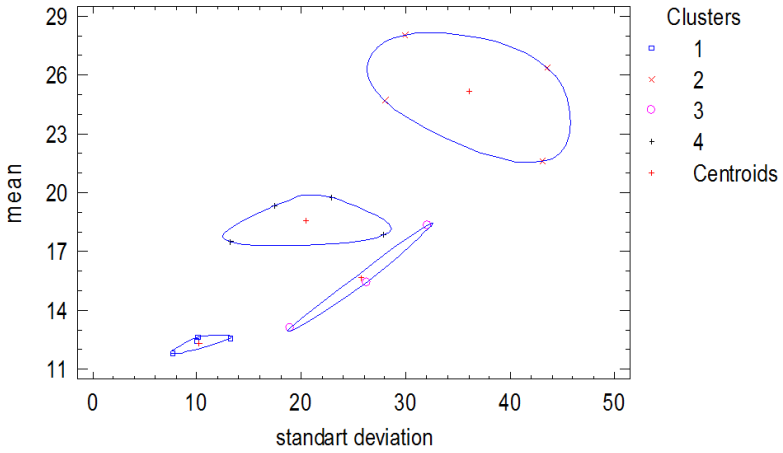


Fig. 3. Cluster analysis scatter plot

distribution parameters - $\alpha = 8.2932$, $\beta = 0.490662$; the fourth subgroup - p-value is 0.29225, distribution parameters $\alpha = 12.97$, $\beta = 0.464609$.

Figure 4 shows a graph of the theoretical density distribution function and histogram. The data are resulted from the clustering process for the first subgroup. Good correspondence between theoretical density distribution function and observed data is obvious.

Further let us consider this analysis in the same way that was described above, but for other groups. The groups were the following: two groups in the humanities took the test consisting of the same questions and the same time limit that the test described above. Restrictions for the test for three groups in the technical discipline were the following: test duration of 30 minutes and 15 test questions. The test for three groups in the natural science discipline had the following restrictions: test duration of 45 minutes and 22 test questions. The number of observations was varied from 64 to 161 for the test on the humanities, from 15 to 120 for the technical discipline, from 26 to 155 for the natural science discipline. The analysis showed that data was described by log-logistic distribution. Only 2.5 % of the observations did not fit the log-logistic distribution. Out of the remaining observations p-value exceeded 0.2 in 97% of cases. It also turned out that the parameters of the log-logistic distribution for the natural sciences and the technical disciplines exceeded the respective parameter for the humanities discipline. One observation from the humanities did not fit the log-logistic distribution due to bimodal data.

Then the data were clustered. On the first step of clustering, several similar observations were grouped into subsets. The Parameters of distribution were used as the objects for the cluster analysis. The Ward method of the cluster analysis was used in this step. As a result, four subgroups were obtained. On the second clustering step we applied the k-means algorithm to the cluster subgroups. Figure 5 shows a scatter plot of the two clustering parameters: parameter alpha and parameter beta.

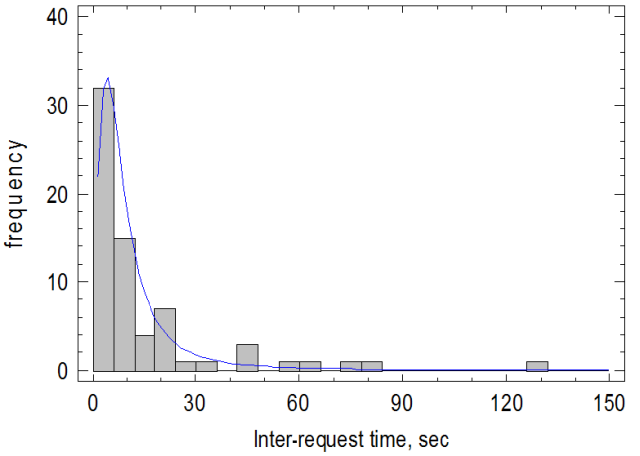


Fig. 4. Log-logistic distribution for the first subgroup

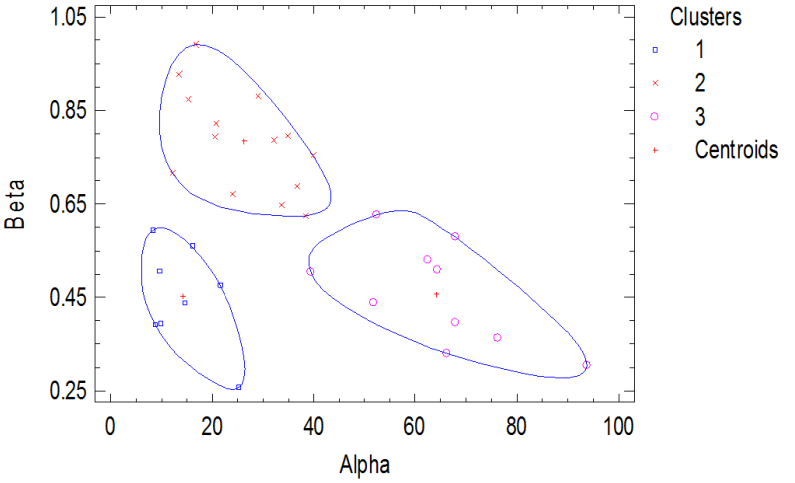


Fig. 5. Two-step cluster analysis

Table 2. The clustering results of the distribution parameters

Cluster number	parameter	parameter
1	14.2718	0.452209
2	26.2297	0.783483
3	64.1761	0.458037

As a result of the analysis we obtained three clusters. Only the humanities discipline belongs to the first cluster. Six distribution parameters characterizing the technical discipline and eight distribution parameters characterizing the natural sciences belong to the second cluster. Six parameters characterizing tests on the technical discipline and four parameters characterizing tests on the natural sciences belong to the third cluster. Table 2 shows centroids of the obtained cluster.

4 Conclusion

After conducting this research we have come to the following conclusion. All students in each group are significantly heterogeneous. Each student has its own strategy for taking the test session, which is one of the reasons for heterogeneity of the group. We consider that there is not enough data to apply limiting laws of probability theory here and it would be a mistake to use the normal distribution in this case.

The inter-request time for a single user is widely varied and can be significantly larger than average inter-request time. It can be used for determination of the test time.

After applying the cluster analysis the number of homogeneous subgroups is varied from 4 to 12.

Among several probability distributions we could find that the log-logistic probability distribution was the best model for observation data. Although the log-normal distribution was good for our data but p-values obtained from K-S test showed the priority of the log-logistic distribution.

Performed analysis proves that the distribution parameter depends on the type of discipline. This parameter is minimal for the humanities and increases for the technical and natural discipline.

References

1. Hollander, M., Wolfe, D.A.: Nonparametric statistical methods. John Wiley & Sons, NY (1999)
2. Zhao, G., Shan, Q., Xiao, S., Xu, C.: Modeling web browsing on mobile internet. IEEE Communications Letters 15, 1081–1083 (2011)
3. Downey, A.B.: Lognormal and Pareto distributions in the Internet. Computer Communications 28, 790–801 (2005)
4. Shuai, L., Xie, G., Yang, J.: Characterization of HTTP behavior on access networks in Web 2.0. In: International Conference on Telecommunications, ICT 2008, pp. 1–6. IEEE (2008)
5. Van der Linden, W.J.: A lognormal model for response times on test items. Journal of Educational and Behavioral Statistics 31, 181–204 (2006)
6. Kvam, P.H., Vidakovic, B.: Nonparametric statistics with applications to science and engineering. John Wiley & Sons (2007)
7. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. John Wiley & Sons (2005)
8. Albov, N.Y., Adzhemov, A.S.: Statistical analysis of incoming traffic of applications for testing services of information and education resources. Elektrosvjaz' 4, 28–30 (2012) (in Russian)

The Role of Enterprise Social Networking (ESN) on Business: Five Effective Recommendations for ESN

Saeed M. Alqahtani, Sultan Alanazi, and Derek McAuley

University of Nottingham, School of Computer Science
{psxsa22,psxsa16,derek.mcauley}@nottingham.ac.uk

Abstract. The growth of social networking over the recent past has been phenomenon and business gains and opportunities generated by social networking sites such as Facebook, MySpace, LinkedIn and Tweeter among others, indicate the potential of social networking to help achieve business outcomes and enhance the conduciveness of contemporary working environments. The challenge of generating the gains and opportunities presented by social networking into the enterprises lies in developing effective frameworks and strategies that are essential in facilitating enterprise social networking (ENS) successfully. The aim of this study is to analyze and investigate the spread of social networking in the workplace in order to determine how a social network influences business and society, as well as how we can reshape and/or guide the ENS to a more efficient future business environment. Then, recommendation systems for an ESN system will be proposed.

Keywords: ESN, Social Networking, Business, Recommendation Systems.

1 Introduction

Social networks have provided a fertile ground to not only socialize with millions of people across the globe on a personal level, but also, it has generated a global platform on which technologically savvy entrepreneurs, enterprises, firms and institutions, be it small or large, private or public and profit or non-profit making to create consumer global awareness of their existence. Moreover, it assists potential and existing customers on what they have to offer, presenting their selling and unique points and having the capacity to penetrate new global markets that would have otherwise been impossible to do or very costly to develop and implement [1]. Primarily, the social media has altered almost all aspects of our lives as indicated by Barlow & Thomas [2].

Social networking in the workplace is a new phenomenon that firms and institutions in the present day that are brave and risk takers are adopting in a bid to capitalize on the strategic benefits. These benefits of social enterprises generate and particularly in their efforts to enhance teamwork, develop healthier interrelations and empower, inspire and encourage its labour forces to be productive, committed,

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_3,

accountable, satisfied in their jobs and more importantly to make them take ownership of the organization's strategic goals and objectives. The social media is such a comprehensive organizational tool, which can easily be used as public relations tool that can be used by the firm or institution to improve their public image and reputation as discussed by Klososky [3].

As noted by Barlow & Thomas, the social networks have evolved the communication landscape, which has eradicated the need for physical meetings that eat into a firm's time and resources that would now be used in enhancing production capacity and has developed virtual offices that allows employees to work anytime and anywhere [2]. According to Klososky, social media offers an element of immediacy, relevance, information flow, connectivity and broadness that lacks in virtually all other social technologies or other forms of media for that matter [3]. Communication in organizational setups has predominantly been one way, with information flowing from the top management to the lower organizational structures and not vice versa but social networking has set in to alter and reassess the hierarchical approach adopted in relation to organizational communication as highlighted by Butler [1]. Despite the great potential and extensive benefits that social networking generates for contemporary firms and institutions, there is a lack of research which has been carried out to analyze and investigate its spread in workplace. This forms the basis of this study, which is to analyze and investigate the spread of social networking in the workplace in order to determine how a social network influences business and society, as well as how one can reshape and/or guide the Enterprise Social Network (ESN) to a more efficient future business environment.

The contribution of this research would help resolve doubts on the effectiveness of social networking to enhance business systems and operations, which help in ensuring the business goals, objectives, mission and vision, are effectively and efficiently achieved. In this study we firstly present a background of ESN influence which covers extensively a literature review on three key concepts. In section (3) we have presented how this study has been investigated. In section (4), (5), we have highlighted in detail the findings, analysis and discussions of the research study by providing the questionnaire analysis, interview analysis, discussion. In section (6), we have listed the most effective recommendation systems for those firms and institutions that seek to implement ESN successfully. Finally, a summary in the conclusion section will be outlined while presenting results further discussing recommendations for future work.

2 Related Work

Modern businesses and institutions are engulfed in stiff business and market competition that is coupled with shifting political, technological, ecological, economical, legal, financial, cultural and social factors that makes the market and business environments unpredictable [4]. Due to advancement in technology, and advancement in communication and transport infrastructures, the modern customer is more informed and therefore, has a high bargaining power than they originally and more and more businesses are coming up, with the two elements combined together causing a

reduced reliable and strong customer loyalty and shrinkage of the market share respectively. This has had a negative impact on the volume of sales, profit margins and stability of majority of firms and institutions [3][4].

For these reasons, modern firms and institutions are finding it crucial to invest in emerging technologies in order to obtain more commitment, collaboration, participation and accountability from their workforces in order to drive up innovation, creativity, teamwork, information and knowledge sharing, which is vital in enhancing a firm's sustainable competitive advantage and increasing quality of production of products and services. Social networking is among technological tools that are being adopted increasingly into workplaces in a bid to develop more efficient, effective, productive and value-added workplaces and businesses [5]. Social networking is an entity that is growing and developing tremendously not only the personal aspects of people, but also, in the systems and structures of business. Social networking in business promises an opportunity to enhance enterprise solutions that are characterized by effective communication between the top management and its labour forces and among employees, collaboration in executing duties and accomplishing of set goals and objectives, improved sharing of information across the board and departments and elimination of barriers to productivity and performance in the workplace [6].

Enterprise social networking is a system that modern firms and instructions cannot effectively succeed, without implementing it [5]. Enterprise social networking just like any other new systems being introduced into existing organisational systems needs careful deliberation into the main goals and objectives of implementing it and the consequences it will bring, an analysis of the impact it will have on the productivity and operations of the business, assessment of its effect on the internal and external environment and evaluating its ability to integrate easily within existing technological infrastructure and how it aligns to the organizational vision, mission, goals and objectives [6][7].

By verifying these important facts, it becomes easier for the organizational management to adapt the appropriate ESN systems, to know when it is appropriate to implement ESN and make available adequate structured and organizational resources required to make ESN establishment successful [8]. Since enterprise social networking is a new phenomenon, there are valid questions on its effectiveness in the workplace, which forms the aim of this study, which is to analyze and investigate the spread of social networking in the workplace in order to determine how a social network influences business and society, as well as how we can reshape and/or guide the enterprise social network to a more efficient future business environment. This next chapter forms the research methodology.

3 Research Methodology

Three approaches have been conducted in this research: Survey, Questionnaires, and Case Study. The research questions for this study includes

1. What are causing the rapid spread of social networking in the workplaces and businesses?

2. What are the negative and positive implications of the increased spread of social networking?
3. What are the impact social networks have on business and the society in general?
4. How can ESN successfully be implemented in the workplace in order to generate effective, efficient and productive enterprise solutions?

By evaluating and measuring the traffic of visitors that visit or frequent a site, it would be help and essential to assess the effectiveness and adequacy of the marketing and advertising strategies used in social networking, which can then be transferred to ESN. A case study on one real company that has implemented ESN and an active social networking site such as Facebook to analyze the trend of visitors on the site, the services they consume has been used. This will help in establishing means of reshaping ESN adopted in workplaces. In this study, there were primary and secondary data, the main data will be collected using questionnaires for the survey, focus group discussions for the case study and online polling. Secondary data will be collected from peer-reviewed journals, textbooks and records of established firms that have implemented ESN. Focus group discussions techniques was appropriate for this study in order to their ability to verify validity of findings collected from quantitative research methods, they allow the researcher to observe the respondents, ask and seek clarifications.

The population samples for this study are respondents aged above 14 years of age from schools, office set-ups, and regular people at public places. The number of respondents will range from a quota of two thousand five hundred people that comprise of 750 employees, who frequently use social networking in their personal lives and not professionally, 250 top managers who frequently use social networking for personal reasons and not for business purposes, 10 line managers working in a real firm that has implemented enterprise social networking and 200 employees from different organizational structures within the real firm that has implemented enterprise social networking, 750 respondents used in the internet polling and 740 respondents of ordinary people at public places. In order to get correct data, each questionnaire will be authorized from the respective agent such as cases of data collected from employees the data will be certified by the manager, in cases of university students the data will be certified by the head of the department, and in cases of school students the teacher will certify the data provided by the student. This is how correct and reliable research can be undertaken. Once the data is collected, there are certain criterions to include and exclude the participant. This research will include how ESN can be improved to help businesses. Therefore young people such as school students are not able to provide the advice in this context; therefore, their response will have low importance as compare to those who are directly linked with the company such as office employees, businessmen, and all other people are directly involved with any business matters.

4 Results and Discussion

4.1 Results

The questionnaire questions were developed based on three key issues of social networking identified in this study. These includes the spread of social networking,

impact of social networking in business and the society and the increased need to transfer social networking into business context by reshaping ESN in order to develop a more efficient future business and workplace climate.

From the data collected from the questionnaires, 99% of the respondents indicated that they logged into their favorite social networking site at least twice a day while the remaining one percent, signed in their social sites at least once a day. When asked how important social networking was in their social lives in a scale of one to five where one was least important, three was fairly important and five was very important, 85% of the responses were that social networking was very important, 13% of the responses were it is fairly important while 2% stated it was least important. As illustrated in figure 1.

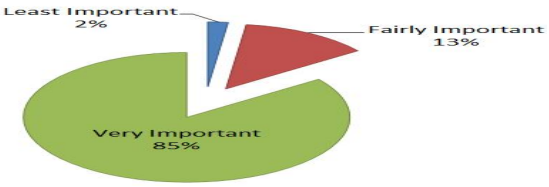


Fig. 1. How Important Social Networking is?

In relation to the widespread use of social networking in social and business environments, respondents were asked to highlight the reasons for the increased spread of social networking. Forty one percent of the responses were that advancement in technology as the greatest catalyst of social networking, 39% indicated that the effectiveness and efficiency of social networking to share, connect, converse, learn, create and entertain was the main reason while 17% stated that the accessibility and availability of social networking sites in varied computer applications was the driving factor. 2% of the responses indicated that the need for people to communicate and socialize often was the underlying cause while 1% indicated that social influence by friends and colleagues to sign up was the reason social networking was spreading so quickly as represented in figure 2.

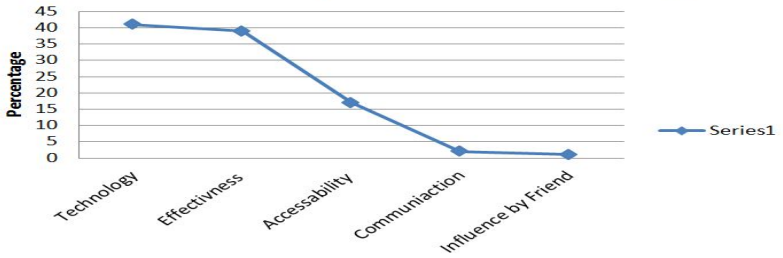


Fig. 2. What Has Caused The Spread of Social Networking In Business and Social Circles?

When queried on the impact social networking had on business and social environments, on a scale of one to three where one represented positive impact, two represented negative impact and three represented neither, 78% said the impact of social networking was positive 20% said it was negative while 2% said that is was neither.

In regards to the questions on what the negative implications of social networking are 71% of the responses indicated that social networking presented security risks such as cracking into other people's accounts and accessibility of vital and essential information by unauthorized users in the business context. 19% indicated that social networking was susceptible to fraud, 6% indicated social networking was prone to cybernetic viral attacks that can damage or cripple intra net systems in workplaces and causes loss of important business documents and records. 4% indicated that it is to blame for thousands break-ups of relationships and marriages as illustrated in figure 3

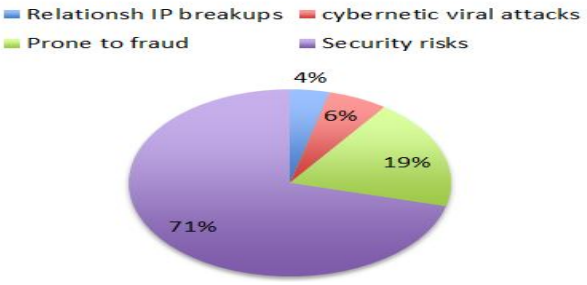


Fig. 3. The Negative Impact of Social Networking

Findings on the positive impact of social networking saw 58% of the participants indicated constant communication and knowledge sharing as the most positive effect of social networking. 22% stated the positive impact entailed connectivity of people from varied cultures and across varied geographical locations and 10% of the respondents indicated that the positive impact constituted to handling of professional processes such as meetings and business transactions from wherever and whenever, making efficient flow of work and communication across organizational units easier and possible.

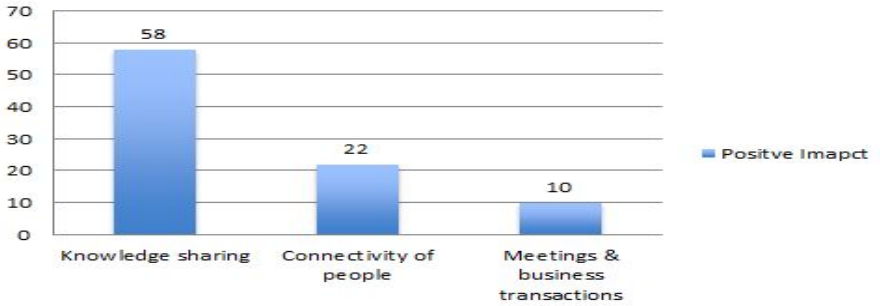


Fig. 4. The Positive Impact of Social Networking

When asked about the social networking sites they knew, 90% of the responses recorded Facebook as their top of mind while 6% recorded Twitter as the top mind and 4% indicated other social networks as their top of mind. Findings on the best ways and means to reshape enterprise social networking in the workplace in a bid to generate effective, efficient and productive enterprise solutions indicate a high percentage. Seventy four percent of the responses were incorporating ESN systems that have familiar features with the commonly used social networking sites such as Twitter and Facebook and making every employee in every organizational level and structure to participate in enterprise social networking. 18% of the responses were effective integration of ESN into existing technological systems and commitment from the entire workforce and the top management. 7% of the responses were setting specific, achievable, measurable and time bound enterprise social networking goals and objectives. One percent of the responses were developing guidelines and measures to guide employees and users in the workplace to ensure they all uphold professionalism require in the business and workplace environment as shown in figure 5.

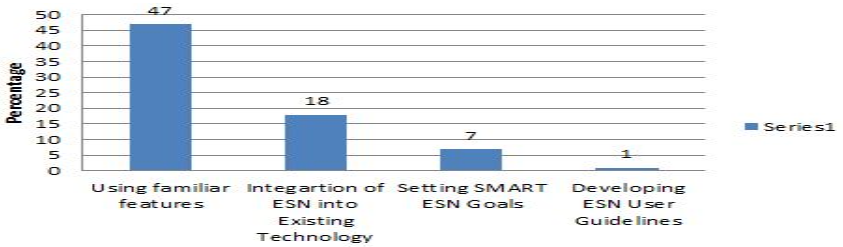


Fig. 5. The Effective Ways of Reshaping Enterprise Social Networking in the Workplace

Eighty percent of responses on what the respondents used social networking sites for stated to connect, share and interact with old and new friends, 15% of the respondents indicated they used them to share business information and meet new business clients while 5% said they used them to meet potential mates.

The second analysis will be on the findings from the focused group discussions. Among ways, organizational leaders are integrating productivity and performance with meeting the social needs of the employees to relate, share and communicate is in ESN. From the findings of the focus group discussions on the most preferred and used social networking used by the respondents, 100% of the responses indicated that all participants preferred and had a Facebook account. Sixty eight percent of the respondents had at least two accounts from Facebook and an additional social networking site where only one account was operational. Twenty percent of the respondents had accounts in three or more social networking sites but only two were active, while 12% of the respondents had at least four accounts in four social networking sites and all of the accounts were operational.

In regards to the spread of social networking in the workplace, 76% of the findings attributed the spread to accessibility and efficiency of social networks to connect people from varied organizational levels at a personal level and the benefits associated with social networking such as identification of unique opportunities and

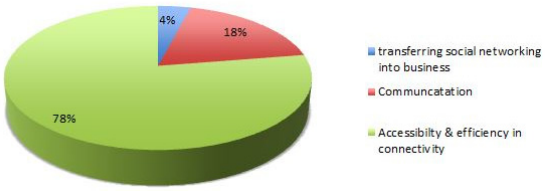


Fig. 6. Reasons for Spread of Social Networking

knowledge sharing. 18% indicated the spread was as a result of the human need to communicate, share and relate constantly while 4% indicated the ease in transfer of social networking systems into the workplace and cost effectiveness of social networks.

Benefits generated by social networking recorded from the focus group discussions amounted to 80% of the responses being elimination of barriers to productivity, increased collaboration and sharing, which generates creativity and innovativeness among workers, 10% of the responses being increased participation by employees and a further 10% indicating the ease in access of valuable and dynamically updated information.

In relation to the best techniques to guide enterprise social networking at the workplace, 80% of the responses were developing effective groundwork and comprehensive ESN implementation strategies and fostering commitment from all stakeholders. Fifteen percent of the responses were development of quantifiable and realistic ESN goals and development of guiding principles and ideals to direct employees on use of ESN professionally. Five percent indicated development of ESN systems with familiar features adopted from commonly used social networking sites used to attract and retain users and fit them into ESN.

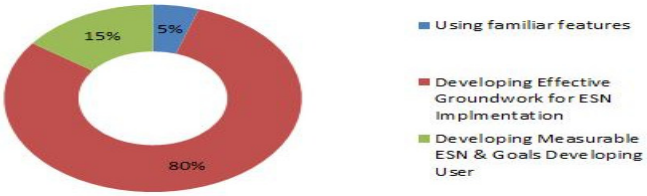


Fig. 7. The Effective Ways of Reshaping ESN

Eight percent of the responses on the questions of negative consequences of social networking at workplaces were time wasting. Ten percent indicating security threats such as accessibility of sensitive information to unauthorized persons and spread of malicious information that can cause emotional and psychological harm to employees, 3% indicating disintegration of strong family and social ties as people are engaged on virtual relationships than real life relationships. 2% indicating spread of security threats such as viruses that can corrupt computer systems at work.

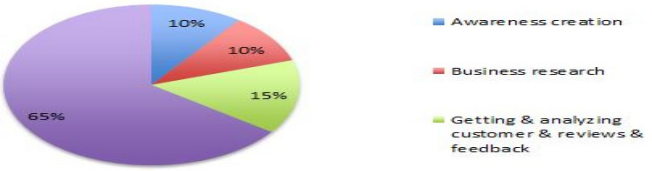


Fig. 8. How Firms Utilize Social Networking in the Workplace

When asked on whether the companies they worked for used social networking for any business reasons 50% of the responses indicated they did while the rest indicted they did not. For those that used social networking in their companies for personal reasons, 65% of them indicated that they used it to promote their products and services to exiting and potential customers. Fifteen percent indicated that they used them to get and analyze reviews and feedback from their customers, 10% indicated they used social networking to conduct business research while 10% indicated they fused them to create awareness about their company, products and services.

4.2 Discussion

The research findings have provided adequate information covering the three key issues identified from the literature reviewed, which are the rapid spreads of social networking, the great impact social networks have on business and the society. In addition, the increased need to transfer social networking into business by reshaping ESN in order to develop a more efficient future business and workplace climate.

As noted from the findings, spread of social networking in the workplace is an accumulation of factors that ranges from its ability to meet the human need to connect, consume and share information among like-minded users. Cost effectiveness in using social networks, increased benefits associated with productivity, engagement, collaboration, knowledge sharing, ability to continually work and operate business issues from anywhere and anytime thus, eradicating the importance of meetings and office settings and the increased ease in access of valuable and dynamically updated information by employees and the top management. However, majority of respondents associated the spread of social networking in the workplace with spread of risks such as viruses, negative publicity caused by spread of malicious rumors as indicated by [9].

Social networking systems fitted effectively in the workplaces helps organizations in tracking the movements of their top performing accounts, analyze and understand the needs of the customer and thus be in a position to produce and deliver quality products and services that effectively and efficiently meet the changing demands, needs and expectations of the customer. This is made possible by posting on-line surveys in ESN sites to examine the attitudes, feedback and views of the customers as highlighted by [10]. the findings indicate that social networking helps develop more productive workforces associated with the employee's ability to share more, know more, collaborate more and know each other from different departments and organizational levels, findings which are supported by [11]. Nevertheless, the findings highlight significant concerns by respondents on the negative impact of social networking

in the workplaces such as time wasting, increased susceptibilities to security risks where vital information falls on wrong hands and malicious rumors and misinformation is spread jeopardizing the reputation of the firm or its workforces. On social level, criminal offenders such as child molesters and sexual offenders have lured their victims using fictitious names and profiles in social networking sites , trapping unsuspecting victims [12]. Nevertheless, David et. al (2012) indicates that implementing ESN from an informed and proactive stance will help limit the risks developed by enterprise social networking and help firms in staying relevant, attracting and retaining sustainable and productive employees [13].

The most significant variable that majority of the respondents indicated as the most effective strategy in reshaping ESN to help develop more efficient business and workplace environments is development of ESN systems with familiar features adopted from social networking sites commonly such as Facebook. Among familiar features that are consistent among social networking sites and can easily fit into ESN systems includes creation of personalized profiles, which are used to find and locate past, current and prospective personal and business contacts. Development of security measures that safeguard users from interactions with persons they do not know or they would not want to connect with, social status, creation of personalized home pages that helps in linkage of like-minded users, features that allow addition, removal of contacts and features that allow users to post comments, comment and share information, videos and pictures.

Findings from the research that employees need to be guided on how and when to use ESN echoes sentiments mentioned in the literature review and supported by [14]. This entails development of policies and regulations guiding employees in the workplace on how they represent themselves as professionals on enterprise social networking, which involves posting business related updates and uploading business appropriate photos, files and videos. This is important because enterprise social networking represents a corporate environment and should be handled as such. As Goodall et al. (2009) suggests, information made available on the enterprise social networking impacts on the perceptions and attitudes of people about the company [15]. This is echoed by Greenleaf (2010) who mentions that effective employees should know what to post in the social networking site and what not to [16] .

5 Recommendations and Conclusion

5.1 Recommendations

This chapter offers effective recommendations for those firms and institutions that seek to implement enterprise social networking successfully and for those that have already implemented and would want to ensure the implementation process remains successful.

5.1.1. Align the ESN Goals with the Goals of the Business and those of Employees

ESN is the wonder drug that organizational leaders having been looking for to cure organizational aliments such as fragmented interrelationship among team members

and across working groups, employee unproductively and reduced employee job satisfaction and limited creativity and innovativeness. These problems have made it difficult for firms and institutions to remain relevant, economically feasible and fail to sustain their competitive advantage. Implementing ESN, which is only successful through full participation by all stakeholders, require that the organizational management to align the ESN goals and objectives to the strategic goals of the business and the employees. Employees are more likely to embrace and make necessarily changes to new organizational systems, if they feel and understand that the new systems would not only help achieve the organizational goals efficiently and effectively, but also, will help in their attaining their personal and professional goals and objectives as supported by [17]. Keeping this in mind, aligning the goals and objectives of employees and the business with the ESN goals and objectives, is the first step in successfully implementing the ESN in the workplace. This is achievable if the organizational leaders understand the needs, expectations and demands of the business and those of its labor forces.

5.1.2. Using Familiar Features from Commonly Used Social Networking Sites

ESN in the workplace is an essential component of organizational communication that has proven effective in eliminating barriers that have traditionally existed between the top management and the workforce. Thus, it will be easy for employees to participate actively in critical organizational processes such as making of decisions, solving operational problems, generating innovative and creative ideas that helps in promoting the competitive advantage of the firm and fostering teamwork, accountability and commitment to ensuring the goals and objectives set are effectively achieved. Therefore, using familiar features is helpful in increasing uptake of the system, enhancing the ability of users to take ownership of the systems and accommodate it and when new systems have features that users are used to, it helps in cutting costs of training them on how to use the new system.

5.1.3. Implement Adequate Security Measures and Frameworks

ESN is characterized by active and effective data sharing anytime and anywhere which generate new challenges of security and confidentiality of information and knowledge being shared. Therefore, there is a greater need to appraise, modify and develop and implement new security policies, procedures, to safeguard shared data against security breach and violation of privacy of information in order to make ESN successful in the business and workplace environments [18]. Developing and implementing effective security measures cannot be overemphasized especially for high security sensitive firms and institutions such as healthcare facilities, the military and research firms among others, who may limit use of ESN due to increased security risks associated with ESN systems. Security policies and systems during implementation of ESN will be fundamental in ensuring information and knowledge shared is not only secure and reliable, but also, available and accurate, which in turn, help disseminate and receive the intended message sent through ESN systems/ sites from users from varied backgrounds and status [19].

5.1.4. Maintain Professionalism When Using ESN Systems

ESN violates the traditional system of a business where the corporate component of the business does not interrelate with the personal component of the stakeholders. Although ESN is more of a socially-engineered concept, the organizational leaders should develop formal structures to guide and remind the labor forces using the ESN Therefore, ESN users should adhere to set ESN regulations, observe legal constraints, understand the intricacies of international property in relation to the messages and information they publish on the ESN sites. Maintaining professionalism on ESN in the workplace ensures that users are still able to adhere to their professional code of ethics and comply with organizational, rules, laws and guidelines, which are important in maintaining order and control over flow of work and operational processes. Additionally, it is a key in sustaining the confidence of the customer in the ability of the firm or institution to produce and deliver professional solutions that they may require.

However, professionalism when using ESN systems is realizable by controlling and managing the type of messages and information, which is shared or posted on the personalized profiles and regulating the type of audios, videos and pictures uploaded on the ESN systems. Moreover, the kind of language used in forums, discussions, and meetings held over enterprise social networking systems. According to Foster et al. (2009), sharing and productivity will improve even more when each employee is accessible from anywhere and anytime [20]. This is achievable by making ESN available in commonly used technological devices such as mobile phones and IPads. In addition, being choosy with what to follow or who to accept invites from, making meaningful real-time feeds and focusing on information, documents and insights that will enhance work performance and productivity.

5.1.5. Commitment

Commitment entails stakeholders within and exterior to the firm being dedicated to ensuring the ESN goals and objectives that are aligned to business strategies and effectively and efficiently achieved. Every successful venture in workplace and business environments rides on the commitment offered by the stakeholders and successful implementation of ESN in the workplace is no different. In implementing ESN, commitment from the top management and all users is vital in making the process viable and successful. Foster et al. (2009) mention that it is important that all participants are devoted to seeing the ESN implementation process succeed regardless of who has initiated the process [20].

5.2 Conclusion and Future Work

From the study, the rapid spread of social networking in the workplace is associated with ease in access of social networking services in cost effective tools such as mobile phones, the core need of people to relate as social beings, an effective means of developing and sustaining contacts with old and new contacts and the immediacy of sending and receiving information over social networking systems. Furthermore, the potential of social networking to contribute to quality life, identify unique

opportunities, linking diverse people from all fronts internationally and the ability to share insights and perspectives, which are essential components of continuous learning, innovation, creativity and enhancing one's ability to respond to change positively.

The positive impact of social networks in the business and the society overrides its negative implications, which only amount to security risks that can be effectively and efficiently be mitigated by reviewing and constant revising of ESN security policies and guidelines. Among the positive impact of social networking verified by the study is the ability to use social networking sites to mobilize and lobby support for social, political and economical agendas, enhancement of the concept of community as people from varied backgrounds globally who have mutual goals, interests, visions and mission are able to start and maintain contact and relate with one another regardless of the distance that separates them and improved communication and knowledge sharing among team members and across different working groups.

What is more, elimination of barriers to productivity, increased collaboration from all quotas of the firm, elimination of boundaries among the workforce in varied organizational structures and enhanced value associated by creativity and innovativeness linked with ESN. By understanding the spread of social networking and the impact it has on business and the society has provided a framework for developing effective strategies for successful implementation of ESN in the workplace in order to develop efficient future workplace and business climates.

As indicated by the research findings and the literature reviewed, effective implementation of ESN is achievable by fostering commitment from all stakeholders in ESN implementation process, developing comprehensive ESN implementation strategies whose goals are aligned to the business goals, developing effective security measures to protect data shared within and between enterprises, maintaining professionalism while using ESN and adopting familiar features commonly used in Social sites into ESN systems. Despite the comprehensive approach adopted by this study, further research are required to investigate additional risks that are generated by ESN in the workplace and how organizations can effectively govern ESN.

References

1. Butler, M., Butler, D.L., Chester, J.: Enterprise Social Networking and Collaboration. Martin Butler Research Limited (2010)
2. Barlow, M., Thomas, D.B.: The executive's guide to enterprise social media strategy: how social networks are radically transforming your business, vol. 42. John Wiley & Sons (2010)
3. Klososky, S.: Enterprise Social Technology: Helping Organizations Harness the Power of Social Media, Social Networking, Social Relevance. Greenleaf Book Group (2011)
4. Meyer, P.: From Workplace to Playspace: innovating, learning and changing through dynamic engagement. Wiley.com (2010)
5. Van Fleet, D.D., Van Fleet, E.W.: The violence volcano: Reducing the threat of workplace violence. IAP (2010)
6. Powell, J.: 33 Million people in the room: How to create, influence, and run a successful business with social networking. Que Publishing (2009)

7. Bernal, J.: Web 2.0 and social networking for the enterprise: Guidelines and examples for implementation and management within your organization. Pearson Education (2009)
8. Newman, A., Thomas, J.: Enterprise 2.0 Implementation. McGraw Hill Professional (2008)
9. Coombs, W.T.: Ongoing crisis communication: Planning, managing, and responding. Sage Publications (2011)
10. Van Winkelen, C., McKenzie, J.: Knowledge Works: The Handbook of Practical Ways to Identify and Solve Common Organizational Problems for Better Performance. Wiley.com (2011)
11. Flemming, L.E.: Reading Keys. Cengage Learning (2013)
12. Lusted, M.A.: Social Networking: MySpace, Facebook, and Twitter. ABDO (2011)
13. David, M., Sutton, C.D.: Social research: The basics. Sage (2004)
14. Butterfield, J.: Professionalism: Soft Skills for a Digital Workplace. Cengage Learning, Singapore (2010)
15. Goodall, S., Goodall Jr, H.L., Schiefelbein, J.: Business and Professional Communication in the Global workplace. Cengage Learning (2009)
16. Greenleaf, C.: The Unwritten Rules of the Workplace: A Guide to Etiquette and Attire for Businessmen. Emerald Book (Distributor), Sidney (2009)
17. Shanks, G., Seddon, P.B., Willcocks, L.: Second-wave enterprise resource planning systems: Implementing for effectiveness. Cambridge University Press (2003)
18. Badke, W.: Research strategies: Finding your way through the information fog. iUniverse (2012)
19. von Hellens, L., Nielsen, S., Beekhuyzen, J.: Qualitative case studies on implementation of enterprise wide systems. IGI Global (2005)
20. Foster, G., Meehan, J., Mason, C., Meehan, K.: Management for social enterprise. Sage Publications (2009)

Dependability and Safety Analysis of ETCS Communication for ERTMS Level 3 Using Performance Statecharts and Analytic Estimation

Tomasz Babczyński and Jan Magott

Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology
Wybrzeże Wyspiańskiego Street 27, 50-370 Wrocław, Poland
{tomasz.babczynski,jan.magott}@pwr.wroc.pl

Abstract. Dependability and safety in European Rail Traffic Management System (ERTMS) level 3 are influenced by quality of localisation of trains and communication. In the paper, dependability and safety of European Train Control System (ETCS) communication for ERTMS level 3 is considered. In ERTMS level 3, traffic is controlled by so called moving block that is associated with the distance between subsequent trains. Relationship between probability of emergency train braking as a function of distance between subsequent trains is studied. In the study, the following parameters are taken into account: train speed, emergency braking distance, train length, position localization error, random variable of emergency braking and stopping times, parameters of transmission between trains and radio block centres, processing time in these centres. In order to find this relationship, the following methods: performance statecharts based and analytic estimation have been proposed. Quality of these methods is investigated.

Keywords: ETCS communication and operation, performance statecharts, Monte-Carlo simulation.

1 Introduction

Dependability and safety parameters in Europe railway domain are specified in standards [5,6]. According to standard [9]: "dependability is the collective term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance". Safety levels are characterized by tolerable hazard rate for Safety Integrity Levels (SIL) [6].

European Rail Traffic Management System (ERTMS) is the European standard for train control and command systems, which is intended to unify all European systems and to enhance cross-border interoperability. European Train Control System (ETCS) is a part of the ERTMS. The ETCS is based on radio communication. There are three levels of ERTMS, with level 3 as the highest. At this level, all important information is exchanged between trains and trackside coordination unit so-called radio block centres (RBCs) via GSM-R (rail GSM communication). A train needs to receive

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_4, © Springer International Publishing Switzerland 2014

so-called movement authorities from the RBC in order to continuously run at high speed. Data processing on board the train and in the RBCs, and radio communication link are crucial factors for safety and efficient operation. Train localization, speed, and integrity information is prepared on board the train, and reported from train to RBC at regular intervals. This enables the RBC to declare the track space behind the train clear which is used in moving block operation mode. It is the main difference when comparing with fixed track blocks used in traditional train traffic.

Dependability and safety are influenced by quality of localization of railway vehicles [4] and communication [7,8,13,14,15]. In ERTMS level 3, train traffic is controlled using moving block that is connected with the distance between two subsequent trains.

The relationship between probability of train stopping as a function of distance between subsequent trains, for given train and ETCS communication parameters, is studied in the paper.

The *problem* investigated in the paper is as follows.
Input (parameters of moving block mode communication and operation):
Train speed,
Emergency braking distance,
Train length,
Position localization error,
Random variable of emergency braking and stopping time,
Probability of incorrect transmission from train to RBC and from RBC to train,
Random variable of duration time of correct transmissions from train to RBC and from RBC to train,
Processing time in RBC.
Output:
Probability of stopping as a function of distance between subsequent trains.

The above problem has been studied in the papers [14,15]. Our study is based on parameters from these papers.

Petri nets [3,11,14,15], and statecharts or state machines [8,10,13,14] are often used in dependability or safety analysis of ERTMS systems. The examples of other modelling languages used in dependability or safety analysis are fault trees and Bayesian networks [7]. Petri nets are formal tool with strong theoretical foundations that are used in expressing the concurrent activities of complex systems. Statecharts or state machines are one of fourteen diagrams of Unified Modelling Language (UML). UML is the standard modelling language in software development. In paper [14], state machines are used as modelling language, while Petri net tool is used for formal verification. In paper [8], application of StoCharts (derived from statecharts) in reliability analysis of train radio communications in ETCS level 3 is presented**.** However, the transformation of the model given in StoCharts into model in MoDeST is required. The last is specification language enriched with facilities to model timed and/or stochastic systems. In paper [12], usefulness of statecharts in safety analysis has been shown.

Our goal is to use one language for modelling and analysis. Therefore, performance statecharts [1] are applied in the paper.

In papers [8,13,14,15], in order to solve dependability or safety problems, simulation experiments are performed or linear equation systems are solved. Performance statechart tool [1] finds the solution by Monte-Carlo simulation.

Additionally, the analytical estimation of the probability of stopping as a function of distance between subsequent trains is given in the paper, and its accuracy is verified. This estimation is expressed by simple formula.

Structure of the paper is as follows. Performance statecharts are recalled in Section 2. Performance statechart model for ETCS communication and operation is given in next section. Analytic estimation of the probability that a train is stopped after emergency breaking is presented in Section 4. For the performance statechart model, this probability obtained by simulation is given in Section 5. In the same section, the estimation of this probability is given too, and estimation accuracy is evaluated. Finally, there are conclusions.

2 Performance Statecharts

The performance statecharts formalism is defined in details in the work [1]. Here, only main syntactic constructs will be characterized. First, a definition of statecharts (without time) will be shown, and then a definition of performance statecharts, which contains the definition of statecharts.

Statechart [1] is a higraph defined as a six tuple:

$$S = <BoxN, childB, typeB, defaultB, ArcN, Arc>, \text{ where:} \quad (1)$$

- $BoxN$ is finite set of state names, which are nodes of the graph depicted as round-edged rectangles named *boxes*.
- $childB \subseteq BoxN \times BoxN$ is hierarchy relation: $<b_1, b_2> \in childB$ means that b_2 is a "child" of "father" b_1. Set $BoxN$ with relation $childB$ defines a tree of states $<BoxN, childB>$. The root r of the tree has no parents, leaves have no childs.
- $typeB : BoxN \rightarrow \{PRIM, XOR, AND\}$ is a function which assign a type to each box. The root r is of type *XOR* (an sequential automaton), the leaves are of type PRIM, and other boxes may be either of type XOR or *AND* (state with orthogonal substates).
- $defaultB : BoxN \rightarrow 2^{BoxN}$ is the partial function that gives the *default* for each box. The default for a XOR box is a set with exactly one box of its children (an initial state of the automaton), while the default for an AND box is the set of all its children. The default for a PRIM box is the empty set.
- $ArcN$ is a finite set of *names* for arcs. $BoxN \cap ArcN = \emptyset$.
- $Arc \subseteq BoxN \times ArcN \times BoxN$ is the set of *arcs*. An arc $\alpha \in Arc$ is a triple $<b_1, a, b_2>$ with $source(\alpha) = b_1$, and $target(\alpha) = b_2$, and $name(\alpha) = a$. It is assumed that arcs are uniquely identified by arc names. Arcs depict transitions between states.

The *performance statechart* PS [1], [2] is a triple:

$$PS = <S, A, L>, \text{ where:} \tag{2}$$

- S is a statechart,
- A is a set of attributes. An attribute has a name and value of type boolean, integer, real or time.
- L is a *labelling function*, which gives an interpretation for arcs. With each transition $\alpha \in Arc$ such that $\alpha = <b_1, a, b_2>$, it associates a label l. The label l is a six tuple:

$$l = < un, pr, te, [gc], al, de >, \text{ where:} \tag{3}$$

  - *un* is a probability distribution function. The function defines opening delay of the associated arc with respect to the time of entry to the source state of the arc provided the arc is opened. That is the earliest time, after which the transition to the destination state can occur.
  - $pr \in Nat = \{1,2,...\}$ is a priority of the arc; the greater number the greater priority of the arc.
  - *te=name(formal_parameters_list)* is an optional trigger event. Its reception by the statechart in the source state makes the transition along the arc *a* eligible to fire, provided its guard condition *gc* is satisfied and the arc is opened. The absence of *te* (depicted with '-') means that the arc may fire immediately after opening. $name \in ExtEVENT \cup IntEVENT$, where *ExtEVENT* is finite set of external events and *IntEVENT* is a finite set of internal events. The first set contains events coming from the environment of the modelled system while the second one includes events generated inside the model.
  - *gc* is a guard condition. It is a Boolean expression that is evaluated when transition is triggered by event *te*, or in absence of *te* in the label, at the each moment, after the arc is opened, such that the values of the attributes used in the expression changes.

  The transition from the source state to the destination state of the arc can fire when:

    - *te* is specified: opening delay time is passed, *te* exists at least after the delay time, guard *gc* evaluates to *true*.
    - *te* is omitted: opening delay time is passed, guard *gc* evaluates to *true* after the arc is opened.

    If more than one transition outgoing from the same state can fire, the one with the highest priority is taken. The transition fires as soon as the above is fulfilled.

  - $al = <a_1,...,a_n>$ is a list of actions. Actions are atomic and executed immediately in the same order they are specified. An action may change valuation of attributes or generate an immediate event. Sets of attributes modified by arcs in orthogonal states has to be disjoined. Generated events are of form *ge=name(actual_parameters_list),* where $name \in IntEVENT$.

- *de* is a list of delayed internal events of the form $<<ge_1, pdf_1>, ..., <ge_n, pdf_n>>$, where $ge_i=name(actual\_parameters\_list)$ is a delayed event, $name \in IntEVENT$, and pdf_i is a probability distribution function, which determines the time of generation of the event ge_i in the future with respect to the time of passing along the given arc.

The opening delay of the arc $\alpha=<b_1,a,b_2>$ is introduced to model some activities within a state b_1 (like *do* actions in UML) before its exiting through the arc *a*. The immediate and delayed events are introduced to model results of actions, which are initiated during passing between boxes through the arc *a*.

3 Performance Statechart Model for ETCS Communication and Operation

We analyse the same scenario of two trains following one of the other on the same rail, as the authors of the papers [14, 15], see Fig. 1. The first train automatically checks its integrity and determines its position. This task, together with the preparing of the *status message*, takes the time of 5[s]. Next, the worked up status message is sent through the GSM-R module to the nearest RBC. Due to the possible transmission errors, the message can be lost with the probability of 1,88%. The successful transmission takes the exponentially distributed random time with the mean value of 450[ms]. Next, the RBC prepares the *authority message* for the succeeding train. This task is 500[ms] long and after it the prepared message is sent using the GSM-R module again (with the same error rate and transmission time as in the previous case). The following train waits for the consequent messages. If the message lacks to come for the long time, the situation becomes hazardous and due to this fact the emergency braking procedure is initiated. After the emergency stop the train is immobilized for the exponential random time with mean value of 15[min] (900[s]).

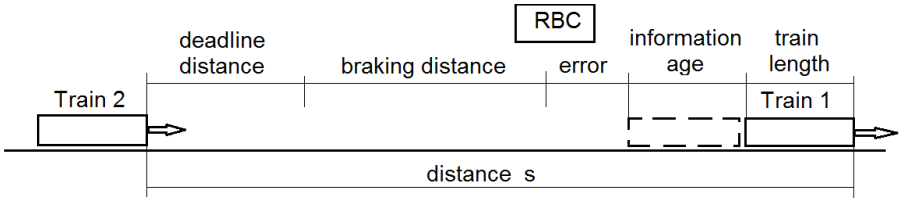


Fig. 1. Distances used in eq. (4)

The deadline used for waiting for the authority message is calculated as the value which ensures the safe stopping of the second train in the case when the first one stops (or lost its integrity) immediately after sending the message that the second train received previously. The deadline is determined from the following equation, (Fig. 1):

$$d = \frac{s-l}{v} - a \tag{4}$$

where s is the initial distance from the head of the first train to the head of the second one, l is the sum of the train length, the braking distance and the measured position error, and is assumed as 3000[m], v is the speed of trains – 300[km/h] (83[m/s]), a is the information age assumed as 5[s] or 9[s] in the future investigations.

The performance model of the system under evaluation was constructed as in paper [14] and is shown in Fig. 2. It is divided into five orthogonal regions. The first two of them model the preceding train: preparing the status message and the communication module. The third one shows the model of the Radio Block Centre – the receiving of the message from the first train, preparing of the authority message to the second train and sending it. The last two orthogonal regions are modelling the receiving of the authority message by the second train and the emergency breaking after the deadline exhausted.

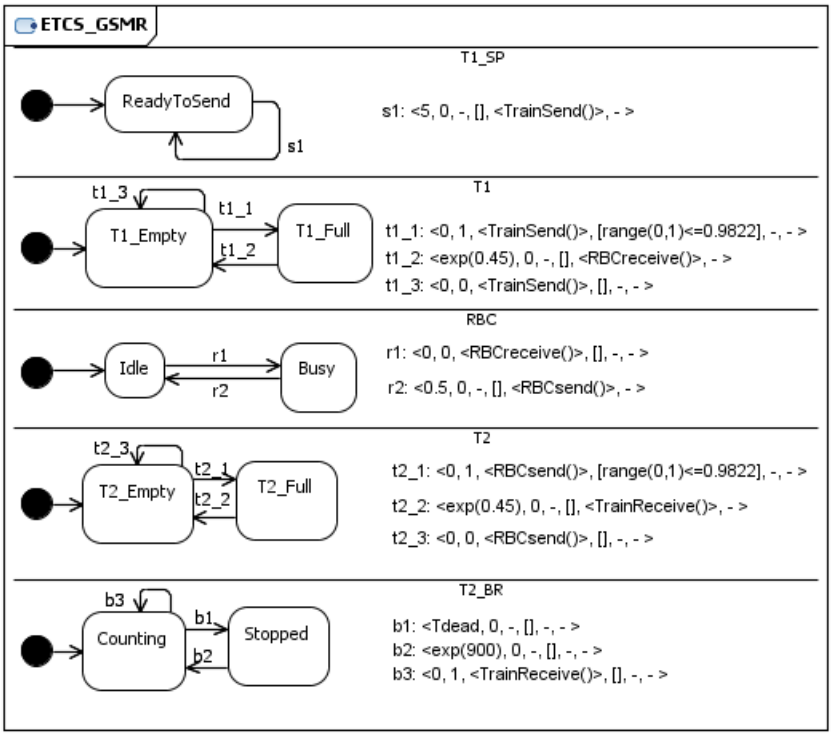


Fig. 2. Performance model of two trains and RBC system

The model was simulated using the Monte-Carlo method. Results are shown in section 5. For small values of the s distance (from the eq. (4)) – less than 5,5[km], the simulation time was very short or short (second parts to minutes) but for greater values it became inacceptable long (days and even more). It leads to the conclusion that the simulator must use any kind of rare event optimisation to successfully evaluate this model.

4 Analytic Estimation for ETCS Communication and Operation

There are two sources of the analytic estimation, namely, performance statechart model in Fig. 2 and formula for deadline *d* given by exp. (4).

Position/integrity package is generated by *Train* 1 each 5[s]. This time will be denoted by p. Let us consider time intervals of length *d*, where *d* has been defined in previous section. For *Train* 1 this interval is started immediately after sending the last package, while for *Train* 2 - after receiving the last package. Some packages that have been sent from *Train* 1 do not reach *Train* 2. The number of the packages generated during the interval for *Train* 1 is

$$n = int\left(d/p\right), \tag{5}$$

where $int(x)$ is the integer part of x. Package transmission times from *Train 1* to the RBC and from the RBC to *Train 2*, and package processing time by the RBC do not influence the mean time between receiving instants of two subsequent packages that should arrive at *Train 2*. Hence, the mean time between these instants is equal to p. It is crucial point in the estimation method.

Let the probability that package transmission from *Train 1* to the RBC is incorrect be q. In considered example $q = 0.0188$. Let the probability that package transmission from the RBC to *Train 2* is incorrect be the same. Package transmission from *Train1* to *Train 2* is incorrect in three cases:

- The package transmission from *Train 1* to the RBC is incorrect, while the package transmission from the RBC to *Train 2* is correct,
- The package transmission from *Train 1* to the RBC is correct, while the package transmission from the RBC to *Train 2* is incorrect,
- The above both transmissions are incorrect.

Hence, the probability that the package transmission from *Train 1* to the RBC is incorrect or that the package transmission from the RBC to *Train 2* is incorrect is equal to:

$$Q = q(1-q) + (1-q)q + q^2 = 2q - q^2. \tag{6}$$

It is the probability of incorrect transmission from *Train 1* to *Train 2*.

The probability that each of n packages that are sent in time interval $(0, d]$ are incorrectly transmitted is $E = Q^n$. It is probability for a train being stopped because of waiting time for the package is longer than the deadline *d*.

One is interested in probability of occurrence of the train being stopped event in one year period. For one train, provided it works one year, the number of package transmissions between *Train 1* and *Train 2* in this period is

$$k = 3600\left[\frac{s}{hour}\right]/p \cdot 24\left[\frac{hour}{day}\right] \cdot 365\left[\frac{day}{year}\right]. \tag{7}$$

Mean number of occurrences of the train being stopped event in one year period is

$$S = Q^n \cdot k\,. \tag{8}$$

Let *st* be the mean stopping time for one stopping. In our example $st = 900[s]$. Mean time of total stopping time in one year is

$$T = st \cdot S. \tag{9}$$

The estimate of probability that a train is stopped is given by:

$$PS = {}^{T}/_{1[year]}. \tag{10}$$

Therefore, after transformations the probability estimate that a train is stopped is:

$$PS = st \cdot \frac{Q^n}{p}. \tag{11}$$

This estimate is not good for small values of *d*. In next section, accuracy of it will be evaluated.

Sometimes, simulation of systems with rare events does not give sufficiently exact results because of time limitations. The above analysis can be done even for such system parameters that some events occur very rarely.

5 Monte-Carlo Simulation and Analytic Estimation Results

In Fig. 3 and 4, probability of stopping as a function of distance between subsequent trains is given.

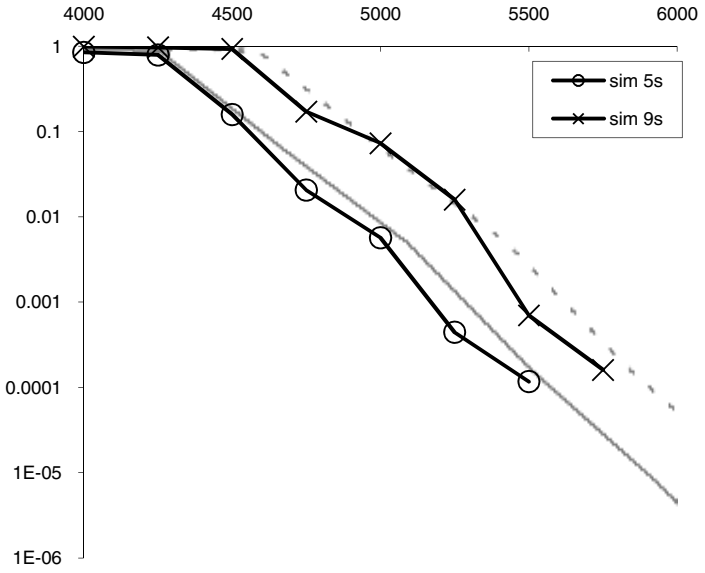


Fig. 3. Simulation results (bold lines) on the background of the chart from [14] (the continuous grey line is for the information age 5[s], the dotted line – 9[s])

In the Fig. 3 the simulation results are shown against the results of simulations performed by the authors of the paper [14]. One can see that both results are very similar. Precise calculating of differences between them is difficult due to unknown parameters of compared simulation e.g. its statistical errors and unknown simulation points.

Fig. 4 shows results of approximations described in Section 4. Each plateau of the stair-shaped line conforms to the same number of messages sent during the deadline time. Again, the very good accuracy can be seen. For the left end of each plateau, the error is bigger. This is the result of ignoring the situation when some of messages transmitted are lost while the other are not lost but simply comes too late. Resolving of this inaccuracy requires deeper investigations.

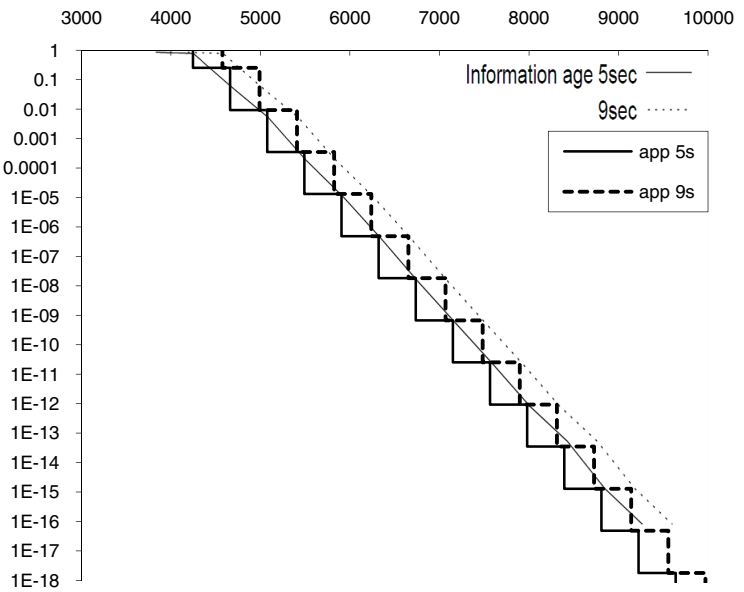


Fig. 4. Approximation results (stair shape lines) on the background of the chart from [14]

6 Summary

Performance statechart model for ETCS communication and operation has been presented. The model is composed of five statecharts that work concurrently. For the model, the probability that a train is stopped after emergency braking as a function of distance between trains has been obtained by simulation. This method is useful for distance between trains smaller than 5.5 km. In order to obtain the above relationship for greater distance between trains, rare events oriented extension is required. Analytic estimation of this probability vs distance between trains has been constructed too. Accuracy of this estimation for distance between trains greater than 5 km is very good. Hence, intervals where our two methods work correctly are joined, i.e. they both cover entire interval of the distance between trains values. Our methods give the relationship in either small simulation or very small analytic estimation computation time.

References

1. Babczyński, T.: Guarded performance statecharts. In: Zamojski, W. (ed.) Inżynieria Komputerowa, pp. 89–103. WKiŁ, Warszawa (2005)
2. Babczyński, T., Kruczkiewicz, Z., Magott, J.: Performance evaluation of multiagent information retrieval system. Foundations of Computing and Decision Sciences 29(1-2), 7–23 (2004)
3. Barger, P., Schon, W., Bouali, M.: A study of railway ERTMS safety with Colored Petri Nets. In: Proc. European Safety and Reliability Conference, ESREL 2009, vol. 2, pp. 1303–1309 (2009)
4. Beugin, J., Marais, J.: Simulation-based evaluation of dependability and safety properties of satellite technologies for railway localization. Transportation Research Part C: Emerging Technologies 22, 42–57 (2012)
5. EN 50126-1,2, Railway applications specification and demonstration of reliability, availability, maintainability and safety (RAMS), Part 1, 2, CENELEC European standard (2000)
6. EN 50129, Railway applications communication, signaling and processing safety related electronic systems for signaling, CENELEC European standard (2003)
7. Flammini, F., Marrone, S., Mazzocca, N., Vittorini, V.: Modeling system reliability aspects of ERTMS/ETCS by Fault Trees and Bayesian Networks. In: Proc. European Safety and Reliability Conference, ESREL 2006, pp. 2675–2683 (2006)
8. Hermanns, H., Jansen, D.N., Usenko, Y.S.: From StoCharts to MoDEST: a comparative reliability analysis of train radio communications. In: Proc. 5th Int. Workshop on Software and Performance, WOSP 2005, Palma de Mallorca, Spain, July 12-14 (2005)
9. IEC 60050-191: Int. El. Vocabulary, Chapter 191 – Dependability and Quality of Service (1990)
10. Magott, J., Skrobanek, P.: Timing analysis of safety properties using fault trees with time dependencies and timed state-charts. Reliability Engineering and System Safety 97(1), 14–26 (2012)
11. Meyer zu Hörste, M., Schnieder, E.: Formal Modelling and Simulation of Train Control Systems Using Petri Nets. In: Wing, J.M., Woodcock, J., Davies, J. (eds.) FM 1999. LNCS, vol. 1709, p. 1867. Springer, Heidelberg (1999)
12. Pap, Z., Majzik, I., Pataricza, A., Szegi, A.: Methods of checking general safety criteria in UML statechart specifications. Reliability Engineering and System Safety 87(1), 89–107 (2005)
13. Qiu, S., Sallak, M., Schon, W., Cherfi, Z.: Modelisation et evaluation de la disponibilite d'un systeme de signalisation ferroviaire ERTMS niveau 2. In: Proc. QUALITA 2013, Compiegne (2013)
14. Trowitzsch, J., Zimmermann, A.: Using UML State Machines and Petri Nets for the Quantitative Investigation of ETCS. In: Proc. 1st Int. Conf. on Performance Evaluation Methodologies and Tools. ACM Int. Conf. Proc. Series, vol. 180, pp. 34–41 (2006)
15. Zimmermann, A., Hommel, G.: Towards modeling and evaluation of ETCS real-time communication and operation. The Journal of Systems and Software 77, 47–54 (2005)

Entropy-Based Internet Traffic Anomaly Detection: A Case Study

Przemysław Bereziński, Józef Pawelec, Marek Małowidzki, and Rafał Piotrowski

Military Communication Institute, Zegrze, Poland
{p.berezinski,j.pawelec,m.malowidzkim,r.piotrowski}@wil.waw.pl

Abstract. Recently, entropy measures have shown a significant promise in detecting diverse set of network anomalies. While many different forms of entropy exist, only a few have been studied in the context of network anomaly detection. In the paper, results of our case study on entropy-based IP traffic anomaly detection are prestented[1]. Besides the well-known Shannon approach and counter-based methods, variants of Tsallis and Renyi entropies combined with a set of feature distributions were employed to study their performance using a number of representative attack traces. Results suggest that parameterized entropies with a set of correctly selected feature distributions perform better than the traditional approach based on the Shannon entropy and counter-based methods.

Keywords: anomaly detection, entropy, netflow, network traffic measurements.

1 Introduction

As the number of network security incidents grows each year [1], network intrusion detection becomes a crucial area of research. Widely used security solutions like firewalls, antivirus software and intrusion detection systems do not provide sufficient protection anymore because they do not cope with evasion techniques and not known yet (0-day) attacks. To cover this area, anomaly detection is a possible solution. Network anomalies may potentially indicate malicious activities such as worms propagation, scans, botnets communication, Denial of Service attacks, etc. The problem of a generic anomaly detection method for network anomalies is still unsolved. Recently, entropy measures have shown a significant promise in detecting diverse set of network anomalies [2-6].

Anomaly may be defined as a deviation from a norm and something which is outside the usual range of variations. Usually, in anomaly detection, a model describing normal circumstances is prepared first, then predictions based on the model are compared with actual measurements. A comprehensive survey about anomaly detection methods has been presented by Chandola et al. in [7]. There are

[1] This work has been co-financed by the Polish National Centre of Research and Development under grant no. PBS1/A3/14/2012 (SECOR - Sensor Data Correlation Engine for Attack Detection and Support of Decision Process).

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_5,

many problems with anomaly detectors which have to be addressed. The main challenges are: high false positive rates, long computation time, tuning and calibration and root-cause identification [7-8].

In our previous work [9], some generalizations of entropy were described in details and some ideas about their possible use for network anomaly detection were presented. In this paper, we make two contributions. First, we present some not commonly known theory regarding entropies used in the context of anomaly detection. Second, we present results of our case study on entropy-based IP traffic anomaly detection that involved a number of entropy variants and a set of different feature distributions.

The paper is organized as follows: First, we discuss related work and overview different form of entropies. Then, we switch to flows and traces, describing a dataset and anomalies it contains. Next, we discuss the applied methodology. In the following sections, we analyze our results. We finish the paper with conclusions and proposals for future work.

2 Related Work

Entropy-based approach for network anomaly detection has been of a great interest recently. This approach relies on traffic feature distributions. Feature distributions give a different view of a network activity than traditional counter-based volume metrics (like flow, packet, byte counts), which are widely used in commercial solutions. Several traffic features, i.e., header-based (addresses, ports, flags), size-based (IP or port specific percentage of flows, packets and octets) and behavior-based (in/out connections), have been suggested in the past [2],[5]. However, it is unclear which features should be used. As an example, Nychis in [2] claims that there is a strong correlation between addresses and ports and recommends the use of size and behavior-based features. On the contrary, Tellenbach in [5] reported no correlation among header-based features. A possible explanation of these contradictory results could be different data sets or, perhaps, some change in Internet traffic characteristics. We propose another possible explanation in section 7.

Although entropy is a prominent way of capturing important characteristics of distributions in a compact form (a single number), some other summarization techniques are proposed in the literature, i.e., histograms [11] and sketches [12]. Their main problem is however the proper tuning.

According to the literature, entropy of feature distributions performs better than widely used counter-based features (like flows, packets and byte counts) [15]. Volume based detection handles large traffic changes (such as bandwidth flooding attacks) well, but a large class of anomalies does not cause detectable changes of volume. Moreover, Brauckhoff et al. in [10] prove that an entropy-based approach performs better than a volumetric one in case sampled[2] flows are used.

Entropy-based methods use the Shannon entropy [2],[15], the Titchener entropy (T-Entropy) [6], and the parameterized Renyi [3] or Tsallis [4-5] entropy. Most authors agree that there are some limitations of entropy-based detection, especially when it

[2] Many routers form flow statistics from a sampled stream of packets in order to limit consumption of resources for measurement operations.

comes to detecting small or slow attacks. This is especially true for the Shannon entropy, which has a limited descriptive power. According to the literature, the range of detectable anomalies for parameterized entropies is wider [5].

3 Entropy

In this section we present some not commonly known theory regarding entropies used in the context of anomaly detection.

Definition of entropy as a measure of disorder comes from thermodynamics and was proposed in the early 1850s by Rudolf Clausius. The statistical definition of entropy as a measure of uncertainty was developed by Ludwig Boltzmann in the 1870s. In 1948, Claude Shannon adopted entropy to information theory. We will start our quick survey with the Shannon's variant, as it is probably the most popular and commonly used entropy.

For a probability distribution $p(X = x_i)$ of a discrete random variable X, the Shannon entropy is defined as:

$$H_S(X) = \sum_{i=1}^{n} p(x_i) log_a \frac{1}{p(x_i)} \quad (1)$$

X is the feature that can take values $\{x_1 ... x_n\}$ and $p(x_i)$ is the probability mass function of outcome x_i. Depending on the base of the logarithm, different units are used: *bits* (a=2), nats (a=e) or hurtleys (a=10). For the purpose of anomaly detection, sampled probabilities estimated from a number of occurrences of x_i in a time window t are typically used. The value of entropy depends on randomness (it attains maximum when probability $p(x_i)$ for every x_i is equal) but also on the value of n. In order to measure randomness only, some normalized forms can be employed. For example, an entropy value can be divided by n or by maximum entropy defined as $log_a(n)$.

Sometimes not only the degree of uncertainty is important but also the extent of changes between assumed and observed distributions, respectively denoted as q and p. A relative entropy, also known as the Kullback-Leibler divergence, may be employed to measure the size of change:

$$D_{KL}(p||q) = \sum_{i=1}^{n} p(i) log_a \frac{p(i)}{q(i)} \quad (2)$$

The Shannon entropy assumes a tradeoff between contributions from the main mass of the distribution and the tail. To control this tradeoff, two parameterized Shannon entropy generalizations were proposed, respectively, by Renyi (1970s) [16] and Tsallis (late 1980s) [17]. If the parameter denoted as α has a positive value, it exposes the main mass (the concentration of events that occur often), if the value is negative – it refers to the tail (the dispersion caused by seldom events). Both parameterized entropies derive from the Kolmogorov-Nagumo generalization of an average:

$$\langle X \rangle_\varphi = \varphi^{-1}(\textstyle\sum_{i=1}^{n} p(x_i)\, \varphi(x_i)) \quad (3),$$

where φ is a function which satisfies the postulate of additivity and φ^{-1} is the inverse function. Renyi proposed the following function φ:

$$\varphi(x_i) = 2^{(1-\alpha)x_i} \tag{4}$$

After transformations, Renyi may be given in the following form:

$$H_{R\alpha}(X) = \frac{1}{1-\alpha} log_a \left(\sum_{i=1}^{n} p(x_i)^{\alpha} \right) \tag{5}$$

Tsallis extended the Renyi entropy with the following function φ:

$$\varphi(x_i) = \frac{2^{(1-\alpha)x_i} - 1}{1-\alpha} \tag{6}$$

After transformations, the Tsallis entropy will be given by:

$$H_{T\alpha}(X) = \frac{1}{1-\alpha} \left(\sum_{i=1}^{n} p(x_i)^{\alpha} - 1 \right) \tag{7}$$

The normalized form of the Tsallis entropy [18] is typically defined as:

$$H_{TN\alpha}(X) = \frac{1}{1-\alpha} \left(1 - \frac{1}{\sum_{i=1}^{n} p(x_i)^{\alpha}} \right) \tag{8}$$

For both the Tsallis and Renyi entropies, parameter α exposes concentration for $\alpha > 1$ and dispersion for $\alpha < 1$. For $\alpha \to 1$, both converge to the Shannon entropy.

Another form used in the context of anomaly detection is the Titchener entropy (T-entropy) [19]. T-entropy is the gradient of linearized form of a string complexity measure called T-complexity. String complexity is a minimum number of steps required to construct a given string. As we mentioned earlier, in typical entropy-based detection, frequencies for values of discreet random variables are used to estimate probabilities. The probabilities must not depend on the occurrence of previous values. In a complexity-based approach, values are concatenated into a string in a sequence (where order matters). The string is then compressed with some algorithm and the output length is used as an estimate for the complexity; finally, the complexity becomes an estimate for entropy. More details about T-entropy is presented in our previous paper [9] and in Einman's dissertation [6].

After a short review on theory, we now switch to networks. In the following two sections, we will discuss a flow-based network analysis and a dataset we have used in our work.

4 Flow-Based Analysis

There are two approaches to a network traffic analysis, namely packet-based and flow-based. A flow-based approach is becoming more and more popular since it is more scalable in the context of network speed. The concept of network flows was

introduced by Cisco and currently is standardized by the Internet Engineering Task Force (IETF). According to the IETF IPFIX working group [21], "A flow is defined as a set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties." In the simplest form, these properties are source and destination addresses and ports. In this work we focus on flow-based network anomaly detection. Flows may be classified on the basis of several different schemes, i.e., size (elephant and mice), duration (tortoise and dragonfly) and burstiness (alpha and beta traffic) [20]. For example, an elephant is any flow with rate exceeding 1% of the link utilization. Duration of a dragonfly is less than 2 sec, while a tortoise lasts longer than 15 min. According to [20], about 45% of Internet flows are dragonflies and less than 2% are tortoises. Taxonomy of network anomalies as discussed in [14] and [15] is presented in Table 1. The table lists examples of both legitimate and malicious network activity.

Table 1. Network anomaly types according to [14] and [15]

Anomaly Type	Description
α Flows	Unusually large point to point byte transfer
DoS, DDoS, DRDoS	Single-source or distributed (also reflected) Denial of Service Attack which may be volumetric, protocol or application based
Flash Crowd	Burst of traffic to a single destination, from a "typical" distribution of sources
Network/Port Scan	Probes to many destination addresses/ports with a small set of source ports/addresses
Ingress-Shift	Traffic shift from one ingress point to another
Outage	Decrease in traffic due to equipment failures or maintenance
Point to Multi-point	Traffic from single source to many destinations, *e.g.* ,content distribution
Worms	Code propagation by exploiting vulnerable services (special case of a network scan)

5 Dataset

The data we used in our case study have been prepared in the following way: First, we captured (hopefully legitimate) traffic from a medium size corporation network using span ports and open source software - *softflowd* and *nfsen*. Then, we mixed this traffic with a subset of the labeled dataset contributed by Sperrotto et al. [13]. This set is based on data collected from a real honeypot which was running for 6 days. The honeypot featured HTTP, SSH and FTP services. The authors gathered about 14 million malicious[3] flows. From the dataset we extracted flows "responsible" for anomalies listed in Table 2.

[3] All flows from honeypots are malicious by its nature, so there is a need to mix them with legitimate traffic for the purpose of anomaly detection testing. With such custom-made datasets benchmarking is hampered.

Table 2. Selected network anomalies

Attack	Relation address /port	Size	Duration	Flows	Packets	Bytes
SSH BrutteForce (A)	1-1/n-1	S	1 h	750	20K	2,5M
WEB Scan (B)	1-1/n-1	M	14 s	660	7K	0,6M
SSH NetworkScan$_1$ (C)	1-n/n-1	M	2,5 min	15K	30K	1,7M
SSH NetworkScan$_2$ (D)	1-n/n-1	L	7 min	23K	300K	34M

Anomaly A represents a one to one brake-in to the SSH service with a dictionary based attack on username and password. This anomaly is relatively slow and small. Anomaly B represents a typical activity of a web scanner. The volume for this anomaly is a bit higher than for anomaly A but the duration is short. Anomalies C and D are examples of network scans. They are characteristic for network-wide worm propagation via service vulnerabilities. The volume for this anomalies is significantly larger.

We placed our mixed legitimate and anomalous (honeypot) traffic in a relational database. We decided to employ bidirectional flows compliant with RFC 5103 as, according to some works - e.g. [2], unidirectional flows may entail biased results in anomaly detection. This assumption required us to perform some conversion from an unidirectional to a bidirectional form.

6 Methodology

Below, we discuss data processing and selected flavors of entropies and feature distributions.

We analyzed our dataset stored in a relational database. Stored procedures were implemented to capture different feature distributions. The anomalies search area was not limited by any filter (per direction, protocol, etc.). We analyzed flows within fixed (5 min) time windows (with no sliding). Next, the Tsallis or Renyi entropies for positive and negative α values were calculated for distributions listed in Table 3. These distributions are commonly employed in entropy-based analysis except for flows duration which is our proposal.

Table 3. Selected traffic feature distributions

Feature	Probability mass function
src(dst)address(port)	$\frac{\text{number of } x_i \text{ as src(dst) address(port)}}{\text{total number of src(dst) addresses(ports)}}$
flows duration	$\frac{\text{number of flows with } x_i \text{ as duration}}{\text{total number of flows}}$
packets, bytes	$\frac{\text{number of pkts(bytes) with } x_i \text{ as src(dst) address(port)}}{\text{total number of pkts(bytes)}}$
in(out)-degree	$\frac{\text{number of adresses with } x_i \text{ as in(out) degree}}{\text{total number of addresses}}$

The following set of entropies presented in Section 3 was selected:

Table 4. Selected forms of parameterized entropy

Denotation	Formula	Comments
Tsallis$_1$	$H = \frac{1 - \sum_i p_i^\alpha}{1 - \alpha}$	general form
Tsallis$_2$	$H = \frac{1 - \sum_i p_i^\alpha}{(1 - \alpha) \sum_i p_i^\alpha}$	normalization
Renyi	$H = \frac{log_2 \sum_i p_i^\alpha}{1 - \alpha}$	general form
Shannon	$H = \frac{log_2 \sum_i p_i^\alpha}{1 - \alpha}$, $\alpha \to 1$	obtained from Renyi ($\alpha \to 1$)

This selection was based on some promising results with flow-based network anomaly detection as reported by [3],[5]. We believe that T-entropy (which was not selected) is more appropriate for packet-based detection, where the order of packets, e.g., requests and responses from servers, really matters.

During the training phase, a profile was built using time-period specific min and max entropy values computed for every <feature, α> pair. During the detection phase, the observed entropy H_α was compared with the min and max values stored in the profile, according to the following rule:

$$result_\alpha(x_i) = \frac{H_\alpha(x_i) - (min_\alpha - k * min_\alpha)}{(max_\alpha + k * max_\alpha) - (min_\alpha - k * min_\alpha)} \quad (9)$$

$$k - threshold\ margin, k \in \langle 0, 0.3 \rangle$$

According to this rule, threshold exceeding is indicated as abnormal dispersion for values less than zero or abnormal concentration for values higher than one. Abnormal dispersion or concentration for different feature distributions is characteristic for anomalies. For example, during a port scan, a high dispersion in port numbers and high concentration in addresses is observable. Detection is based on the relative value of entropy with respect to the distance between min and max. Coefficient k in the formula (9) determines a margin for min and max boundaries and may be used for a tuning purposes. A high value of k, e.g. k = 0.3, limits the number of false positives while a low value ($k \to 0$) increases detection rate.

For comparison, we also employed a traditional counter-based approach for flow, packet and byte counts. We assumed ideal conditions to measure detection rate for anomalies. We used the same part of legitimate traffic during training and in the detection phase so no false positives could be observed. To measure the false positives rate, we cross-checked legitimate traffic using two halves of the profile.

Finally, linear and rank correlation of entropy timeseries for different α values and different feature distributions was performed in order to define a proper range of α values, and to verify the legitimacy to use a broad spectrum of features. The results are presented in the next section.

7 Results

We obtained the best results for detection of high dispersion and concentration with the $Tsallis_1$ and Renyi entropies. Comparing the $Tsallis_1$ and $Renyi_1$ we observed higher values of threshold excess (more significant peaks of entropy values) with $Tsallis_1$, although such sensitivity was also visible in the form of sharp false positives with legitimate traffic cross check. The Renyi entropy was not so sensitive to anomalies (the excess of threshold was smaller than $Tsallis_1$) but was a bit less vulnerable to false positives. With a traditional counter-based approach anomalies A and B were undetectable by a flow, packet and byte counts, while anomalies C and D were detectable only by flow count. The results for $Tsallis_2$ were ambiguous (slightly exceeding threshold). The Shannon entropy detection failed for anomalies A and B. All results are presented in Table 5. The markings +, +/-, – denote, respectively, successful, indecisive and unsuccessful detection.

Table 5. The effectiveness of selected entropies and volume counters

Attack	$Tsallis_1$	$Tsallis_2$	Renyi	Shannon	Flows	Packets	Bytes
SSH BrutteForce (A)	+	+/-	+	-	-	-	-
WEB Scan (B)	+	+	+	-	-	-	-
SSH NetworkScan$_1$ (C)	+	+/-	+	+	+	-	-
SSH NetworkScan$_2$ (D)	+	+/-	+	+	+	-	-

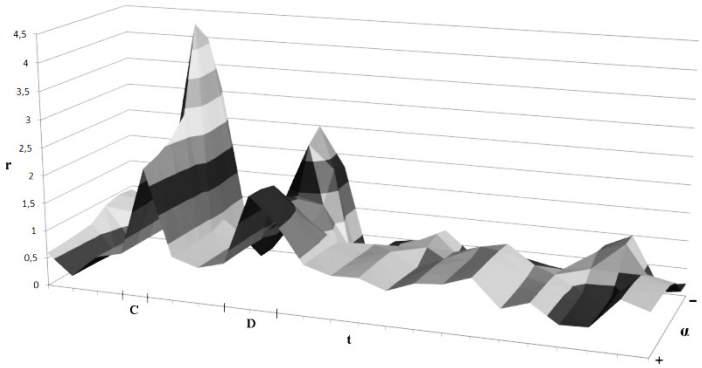


Fig. 1. Abnormally high dispersion in destination addresses for anomalies C and D ($Renyi_1$)

We noticed that with an entropy-based approach some feature distributions work better than others. The best results were obtained by using addresses, ports and flows duration distributions, although we think that the set of proper features is specific for a particular anomaly– thus, a number of different, uncorrelated features (see the second part of the section) should be employed. Abnormally high dispersion in destination addresses distribution for anomalies C and D exposed by negative values of alpha parameters, is depicted in Fig. 1. We can see time *t* on x axis (5 minute time

windows), result *r* (where value above one means abnormal dispersion and below zero means abnormal concentration - formula (9)) on y axis and α values on z axis. Anomaly durations are marked on the time axis.

Abnormal concentration of flows duration for anomaly B - which is typical for anomalies with fixed data stream - is depicted in Fig. 2.

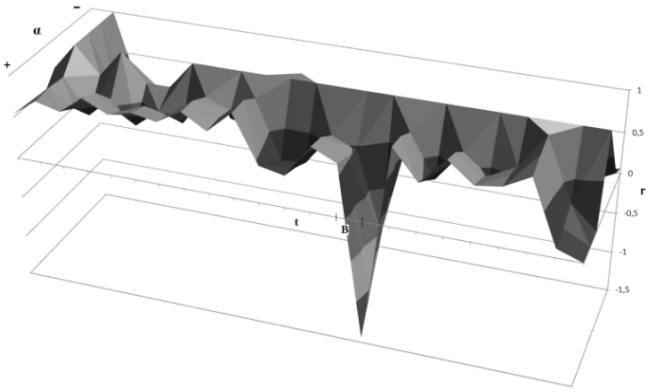


Fig. 2. Abnormally high concentration in flows duration for anomaly B ($Tsallis_1$)

Fig. 3 shows unsuccessful detection (no excess of threshold) of anomaly with common approach based on flow, packet and byte counters.

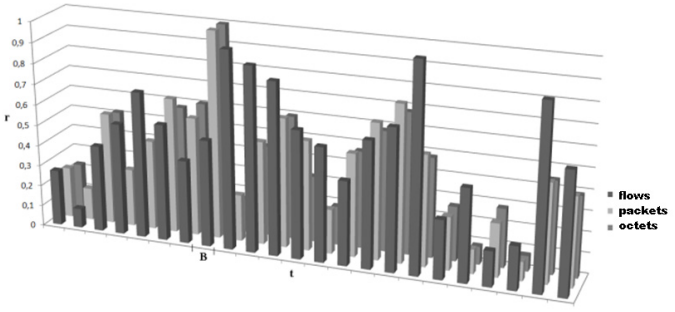


Fig. 3. Unsuccessful detection of anomaly B with a counter-based approach

In the remainder of this section, we analyze correlations for various α values and for various feature distributions. This is important as strong correlation suggests that some results are closely related to each other and thus it may be sufficient to restrict the scope of analysis without impairing its validity.

The results of correlation between entropy timeseries for different α values are presented in Table 6. The table shows the pairwise $Tsallis_1$ α correlation scores from range <-1..1> where scopes |1-0.9|, |0.9-0.7|, |0.7-0.5|, |0.5-0| denote, respectively, strong, medium, weak, and no correlation. The sign determines if the correlation is positive (+/no sign) or negative (-). The presented values (see Table 6) are an average from 15 different feature distributions scores. For the Renyi entropy, the results were similar.

Table 6. Results of linear and rank correlation of α

Pearson	α=-3	α=-2	α=-	α=0	α=1	α=2	α=3
α=-3	1	0,9	0,9	0,66	0	-0,06	-0,09
α=-2	-	1	0,9	0,69	0	-0,06	-0,09
α=-1	-	-	1	0,75	0	-0,05	-0,08
α=0	-	-	-	1	0		0,12
α=1	-	-	-	-	1		0,82
α=2	-	-	-	-	-	1	0,97
α=3	-	-	-	-	-	-	1

Spearman	α=-3	α=-2	α=-	α=0	α=1	α=2	α=3
α=-3	1	0,9	0,8	0,46	0	-0,09	-0,11
α=-2	-	1	0,9	0,57	0	-0,07	-0,1
α=-1	-	-	1	0,72	0	-0,06	-0,09
α=0	-	-	-	1	0	0,2	0,15
α=1	-	-	-	-	1		0,79
α=2	-	-	-	-	-	1	0,98
α=3	-	-	-	-	-	-	1

It should be noticed, that there is a strong positive linear (Pearson) and rank (Spearman) correlation for negative α values and strong positive correlation between α values which are higher than 1. For α = 0 there is some small positive correlation with negative values. For α = 1 (Shannon) there is a medium correlation with α = 2 and α = 3. These results suggest that it is sufficient to use α values from range <-2,2> to have different sensitivity levels of entropy. Some interesting results of pairwise correlation between the $Tsallis_1$ entropy timeseries of different feature distributions are presented in Table 7 and Table 8 (the results for the Renyi entropy were similar).

Table 7. Results of correlation of features for α=-3

Pearson	ip_src	ip_dst	port_src	port_dst	indegree	outdegree
ip_src	1	0,8	0,89	0,91	0,37	0,35
ip_dst	-	1	0,98	0,89	0,27	0,55
port_src	-	-	1	0,86	0,15	0,5
port_dst	-	-	-	1	0,41	0,53
indegree	-	-	-	-	1	0,27
outdegree	-	-	-	-	-	1

Spearman	ip_src	ip_dst	port_src	port_dst	indegree	outdegree
ip_src	1	0,9	0,85	0,87	0,47	0,69
ip_dst	-	1	0,96	0,89	0,43	0,83
port_src	-	-	1	0,83	0,3	0,69
port_dst	-	-	-	1	0,52	0,76
indegree	-	-	-	-	1	0,48
outdegree	-	-	-	-	-	1

Table 8. Results of correlation of features for α=3

Pearson	ip_src	ip_dst	port_src	port_dst	indegree	outdegree
ip_src	1	-	-0,34	-0,02	-0,07	0,44
ip_dst	-	1	-0,29	0,05	0,08	-0,28
port_src	-	-	1	-0,42	0,59	-0,04
port_dst	-	-	-	1	-0,39	0,01
indegree	-	-	-	-	1	0,03
outdegree	-	-	-	-	-	1

Spearman	ip_src	ip_dst	port_src	port_dst	indegree	outdegree
ip_src	1	0,0	-0,21	0,07	0,21	0,366
ip_dst	-	1	-0,31	0,07	0,08	-0,35
port_src	-	-	1	-0,55	0,64	0,23
port_dst	-	-	-	1	-0,53	0,12
indegree	-	-	-	-	1	0,18
outdegree	-	-	-	-	-	1

We presented results for one positive and one negative value of α because these results differ significantly. Averaging (based on results from the whole range of α values) would hide an essential property. It is noticeable that there is a strong positive correlation of addresses and ports for negative values of α but no correlation for positive values. Thus both Nychis [2] and Tellenbach [5] could have been right.

8 Conclusion and Future Work

Concluding the results of our case study, we can observe that:

i) Not normalized Tsallis and Renyi entropies performed best;

ii) The Shannon entropy and counter-based methods performed poorly; in fact, they were unable to detect neither small nor medium-size attacks;

iii) A broad spectrum of features provides a better flexibility to detect different types of anomalies;

iv) Among a large set of network traffic feature distributions, addresses, ports, and flows durations proved in our study to be the best choices.

While we admit that our experiments were limited to a small number of cases, we also believe that the cases were representative. The analyzed dataset contained traces of typical network attacks. Thus, while more research work is necessary to validate the effectiveness of the methods that performed well, the poor performance of the Shannon entropy and counter-based methods allows to question whether they are the right approach to anomaly detection.

The work described in the paper will be the basis for implementation of a sensor cooperating with a multi-source event correlation engine. Thus, we are concerned with a high detection rate rather than a small false positives ratio. Future works will include classification based on results from various features and anomaly details

extraction (e.g., addresses and ports of top contributors to malicious traffic). We are also considering the analysis of additional features. We hope we will be able to report valuable results.

References

1. Verizon Risk Team 'Data Breach Investigations report', Verizon (2012)
2. Nychis, G., et al.: An Empirical Evaluation of Entropy-based Traffic Anomaly Detection. In: ACM SIGCOMM Conference on Internet Measurement (2008)
3. Ruoyu, Y., et al.: Multi-scale entropy and renyi cross entropy based traffic anomaly detection. In: IEEE International Conference on Communication Systems, ICCS (2008)
4. Ziviani, A., et al.: Network Anomaly Detection using Nonextensive Entropy. IEEE Communications Letters 11(12) (2007)
5. Tellenbach, B.: Detection, Classification and Visualization of Anomalies using Generalized Entropy Metrics. Dis. Th., Elektro-Technische Hohschule Zurich (2012)
6. Eimann, R.: Network Event Detection with Entropy Measures. Dis. Th., University of Auckland (2008)
7. Chandola, V., et al.: Anomaly detection: A survey. ACM Comput. Surv. 41(3) (2009)
8. Brauckhoff, D.: Network Traffic anomaly Detection and Classification. Dis. Th., Elektro-Technische Hohschule Zurich (2010)
9. Pawelec, J., et al.: Entropy Measures For Internet Traffic Anomaly Detection. In: TransComp Conference on Computer Systems, Industry and Transport (2013)
10. Brauckhoff, D., et al.: Impact of packet sampling on anomaly detection metrics. In: ACM SIGCOMM Conference on Internet Measurement (2006)
11. Stoecklin, M.P., Le Boudec, J.-Y., Kind, A.: A two-layered anomaly detection technique based on multi-modal flow behavior models. In: Claypool, M., Uhlig, S. (eds.) PAM 2008. LNCS, vol. 4979, pp. 212–221. Springer, Heidelberg (2008)
12. Dimitropoulos, X., et al.: The eternal sunshine of the sketch data structure. Computer Networks 52(17) (2008)
13. Sperotto, A., et al.: A Labeled Data Set For Flow-based Intrusion Detection. In: IEEE International Workshop on IP Operations and Management (IPOM), Berlin (2009)
14. Plonka, D., Barford, P.: Network anomaly confirmation, diagnosis and remediation. In: Allerton Conference on Communication, Control, and Computing. IEEE Press (2009)
15. Lakhina, A., et al.: Mining anomalies using traffic feature distributions. In: ACM SIGCOMM Conference on Internet Measurement (2005)
16. Renyi, A.: Probability Theory. North-Holland, Amsterdam (1970)
17. Tsallis, C.: Possible Generalization of Boltzmann-Gibbs Statistics. Statistical Physics 52(1-2) (1988)
18. Gupta, P., Kumar, V.: General Pseudoadditivity of Kapur's Entropy prescribed by the existence of equilibrium. International Journal of Scientific & Engineering Research 1(3) (2010)
19. Titchener, M.: Deterministic Complexity and Entropy. Fundamenta Informaticae 64(1-4) (2005)
20. Lan, K., Heidemann, J.: On the correlation of Internet flow characteristics. Technical Report ISI-TR-574, USC/Information Sciences Institute (2003)
21. Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, RFC 5101 (2008)

A Formal Approach for Preventive Maintenance Workload Balancing

Ammar Bessam

CIM Maintenance, Brossard (Quebec), Canada
bessamamar@yahoo.fr

Abstract. There is a strong relationship between maintenance planning and production scheduling as both services operate on the same assets and their availability. Many academic research papers and experimental techniques have been developed to synchronize activities between these two inter-dependent key roles in all companies. These techniques are taken into account and implemented in several maintenance management systems to enhance communication and synchronization during the maintenance management process. However, there is no sufficient improvements in the earlier steps of this process. Preventive maintenance workload balancing consists of having a high level abstraction of the asset availability for maintenance work, based on a crossover of maintenance resource availability and operation constraints. It is an earlier task of the maintenance management process. In this paper, we have developed a formal approach for balancing the preventive maintenance workload. This approach allows us to pinpoint the problems and possible irregularities on time in order to obtain the desired behaviors and areas for improvement. The formal workload balancing for preventive maintenance significantly reduces the consumption of necessary resources and provides a good distribution of workload while considering operational constraints.

Keywords: Preventive Maintenance, Workload Balancing, Asset Criticality, PM flexibility, Craft-men hours, Work Priority, CMMS.

1 Introduction

A strong communication exists between preventive maintenance planning and production scheduling. Their activities are inter-dependent because they operate on the same assets and they manage their availability [1]. Many academic and industry works have developed different techniques to enhance the relationship quality and the communication between these two key services in different companies. These techniques are implemented in most of maintenance management systems. There are several systems of maintenance management (e.g. “IBM-MAXIMO”, “CIM-Visual Planner”, …) covering most of the stages of the maintenance management and planning processes. These systems have specific patterns and techniques to handle the majority of maintenance management tasks. However, there is no sufficient specific techniques for balancing the workload of maintenance. Despite the critical role of this task in the optimization of maintenance management from multiple viewpoints.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_6,

Preventive maintenance workload balancing reduces significantly the consumption of crafts resources. Also, it provides an optimized distribution of workload while considering operational constraints. This leads to a better utilization of human resources, reduced paperwork and improved efficiency. At the operational level, the assets will be placed in maintenance at the time when their maintenance does not affect the production.

This paper presents a formal technique to generate multiple scenarios of dispatching workload within the maintenance team. It gives production planners a view of the amount of time required for this preventive maintenance so that they can proactively plan for the release of assets for those periods.

In order to have a complete and an optimized view of the workload balancing, we have based on maintenance management best practices. Also, we have decompiled several maintenance management systems to inspire the right constraints affecting the work balancing process. Our approach is applicable both for initial workload balancing and workload re-balancing. Initial balancing can be performed for a period of one or more years. It is performed at the beginning of this period to prepare the preventive maintenance work scheduling on the enterprise assets. The re-balancing of workload consists of re-balancing of a portion of the workload within the time interval of the initial balancing. This type of balancing is intended to make improvements or corrections result in significant changes in the timing of labors or equipments availability.

Applying our approach guides the maintenance responsible to make the right decisions at the right time. To ensure that each local team has sufficient resources to cover all the work that may arise in their fields, balancing workload according to long-term forecasts of labor must be produced. If the long-term prediction shows that the level of maintenance activity is about to exceed the level that can be accomplished with existing resources, such notice will ensure that there will be enough time to recruit and train additional resources before the situation goes out of control. Similarly, a decrease in the activity level of preventive maintenance to give sufficient time to the possibility of reallocating resources of trades to other teams or activities.

Production planners set to their production schedules the necessary adjustments based on the result of workload balancing. Like this, the equipment will be made available for maintenance at the right time. At this level, this allocation is made to a high level of detail. The exact dates and times for maintenance work will be defined one or two weeks before the deadline. This consists of weekly scheduling.

The remaining of this paper is structured as follows: The first part is reserved for the elicitation of attributes related to preventive maintenance, assets and used maintenance programs. The second point shows the context of the new approach. In the third part, we present the principle of our new approach to balance the preventive maintenance workload. Through the approach presentation, some examples are given to clarify the approach critical nodes.

2 Elicitation of Workload Balancing Criteria

To extract the right attributes and techniques required for a formal workload balancing we have based on two different sources. First, we have analyzed a multitude of

Maintenance Management Systems to elicit different attributes that we have used as a basic data of our approach. Then, we have done a large study of several work management techniques [3], [4], [5], [6] to inspire appropriate data structures and business rules. Below, we present some criteria categories resulted from this analysis. These criteria are in the kernel of our approach:

- Details of related equipment for preventive maintenance ,
- The duration of preventive maintenance work,
- The frequency of preventive maintenance work,
- The next expected maintenance deadlines,
- The work charge (i.e. what will occupy the time of the craft),
- Allowances for meetings / training and overtime,
- The balancing period (ex. : 01/01/13 to 12/31/14)
- The Craft-Men hours,
- The previous year's availability calendars,
- % of the wrench-time, backlog and emergency,
- The flexibility of the preventive maintenance work.

3 The Workload Balancing Approach Context

This flowchart (fig. 1) presents the whole context of our approach. To start the approach, the scheduler should select different criteria to be considered in the workload balancing scenario. After, he defines its own values for each criteria. The goal is to produce several scenarios (according selected criteria and given values). One of these scenarios must be approved before being applied or used.

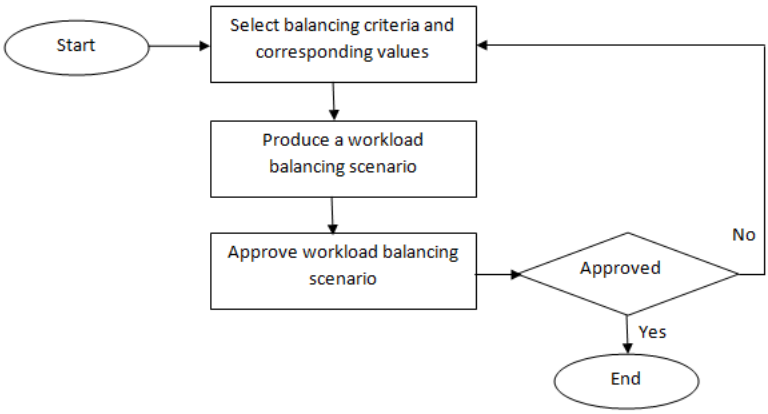


Fig. 1. Approach context diagram

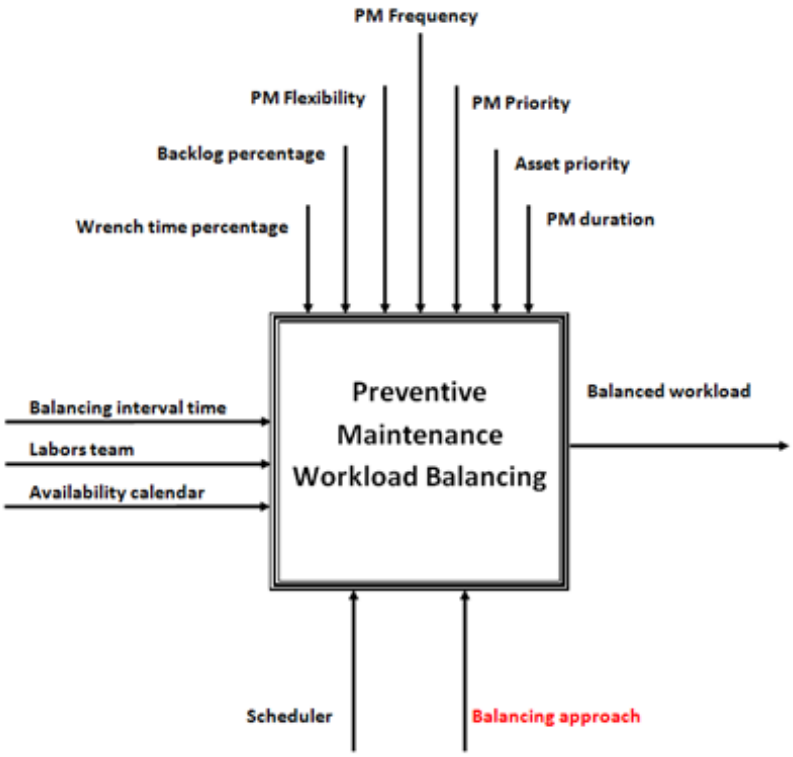


Fig. 2. Context Activity Diagram

The previous SADT (Structured Analysis and Design Technique) based activity diagram [2] shows different required data to produce one scenario of workload balancing.

Once conditions are selected, a scenario will be provided by the tool. When creating the scenario, blocks of PM will be created and spread evenly over the selected period (the number of blocks will represent the frequency of PM).

4 Workload Balancing Approach Principle

4.1 Criteria Definition

Workload balancing operation consists of balancing a charge of preventive work of maintenance on an amount of craft-men hours. It takes into account a set of criteria selected by the actor that will execute the operation. The approach design makes it extensible from multiple viewpoints. Especially, user can include more criteria to obtain more rigorous scenarios. Also, he can ignore one or more basic criteria to have more flexible scenario.

The check box represents either the criteria is selected or deselected. Numbers given for each criteria represent the criteria order when the approach is executed. "Asc or Desc" column is used to specify the order of PMs for each criteria according to their values. The flexibility criteria has the particularity to be mandatory in the first position because we should place non-flexible preventive maintenances to ensure that we still have enough craft-men hours availability.

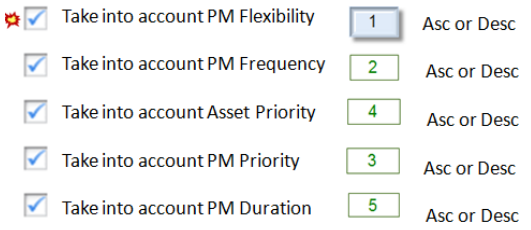


Fig. 3. Criteria definition form

To present how to use different criteria in the approach, we will explain them using the same template. For each criteria, we construct an activity diagram to underline five view points of the corresponding activity:

- **Input Data:** required inputs for each criteria related step;
- **Output Data:** the result of each criteria related step,
- **Activity:** the name of corresponding activity,
- **Control Data:** Required control information to enhance scenario quality,
- **Mechanism:** used resources to execute each activity.

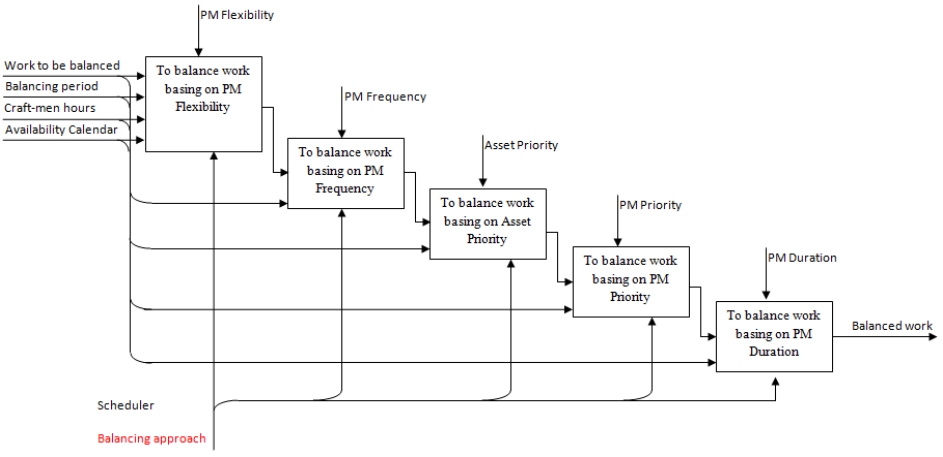


Fig. 4. Activity Diagram: First detail level A1

The global description (presented in fig. 2) is broken-down into a number of sub-activities depending on number of criteria defined by the user. In this case, fig. 4 presents the resulted sub-activities.

- Workload balancing basing on work Flexibility level.
- Workload balancing basing on PM Frequency.
- Workload balancing basing on asset Priority.
- Workload balancing basing on PM Priority.
- Workload balancing basing on PM Duration.

4.2 Work Load Balancing Basing on PM Flexibility

According to workload management, we have to start with inflexible PMs. This allows placing non-flexible work easily in the right period. The rationale is that the non-flexible PMs have production related constraints and availability of assets. So to have a best availability of assets for production, we must be based on this approach. Balancing flexible PMs starts once all non-flexible work is correctly programmed.

Input Data

- A set of Work to be balanced,
- Balancing Period,
- Craft-men hours,
- Availability calendar.

Control Data

- PM flexibility.

Sequences

- Construct a list of non-flexible preventive work.
- Construct a list of flexible preventive work.

First, execute steps presented in sub-section 4.7 for non-flexible preventives maintenances. When all non-flexible work is scheduled, apply the same steps on the flexible preventive work.

Output Data

- PMs are placed,
  OR
  - ✓ List of flexible PMs and a list of non-flexible PMs, and
  - ✓ Remaining Craft-men hours,

4.3 Workload Balancing Basing on PM Frequency

For initial work balancing, we should start by PMs with smaller frequency. In the case of a re-balancing, we must begin by PMs with longer frequency.

Input Data

- Balancing Period,
- Remaining Craft-men hours,
- Availability calendar.

AND

- ✓ A list of flexible and a list of non-flexible PMs, or
- ✓ A list of MPs with the same priority, or
- ✓ A list of PMs with assets they have the same priority, or
- ✓ A list of PM with the same duration.

Control Data

- PM Frequency.

Output Data

- PMs are placed,

OR

- ✓ List of MPs with the same frequency for the classification according to the following criteria, and
- ✓ Remaining Craft-men hours,

4.4 Workload Balancing Basing on Asset Priority

Input Data

- Balancing Period,
- Remaining Craft-men hours,
- Availability calendar.

AND

- ✓ A list of flexible and a list of non-flexible PMs, or
- ✓ A list of MPs with the same priority, or
- ✓ A list of PMs with the same frequency, or
- ✓ A list of PM with the same duration.

Control Data

- Asset Priority.

Output Data

- PMs are placed,
  OR
  - ✓ List of MPs with assets having the same priority for the classification according to the following criteria, and
  - ✓ Remaining Craft-men hours,

4.5 Workload Balancing Basing on PM Priority

Input Data

- Balancing Period,
- Remaining Craft-men hours,
- Availability calendar.

AND

- ✓ A list of flexible and a list of non-flexible PMs, or
- ✓ A list of PMs with the same frequency, or
- ✓ A list of PMs with assets having the same priority, or
- ✓ A list of PM with the same duration.

Control Data

- PM Priority.

Output Data

- PMs are placed,
  OR
  - ✓ List of MPs having the same priority for the classification according to the following criteria, and
  - ✓ Remaining Craft-men hours,

4.6 Workload Balancing Basing on PM Duration

Input Data

- Balancing Period,
- Remaining Craft-men hours,
- Availability calendar.

AND

- ✓ A list of flexible and a list of non-flexible PMs, or

- ✓ A list of PMs with the same frequency, or
- ✓ A list of PM with the same priority.
- ✓ A list of PMs with assets having the same priority.

Control Data

- PM Duration.

Output Data

- PMs are placed,

  OR

  - ✓ List of MPs having the same duration for the classification according to the following criteria, and
  - ✓ Remaining Craft-men hours,

4.7 Sequences

Next steps presents the detail of activities to be executed for each iteration. Each iteration is based on a separate balancing criteria.

- Build lists of PMs according to the corresponding criteria (i.e. decompose the received list to several lists. PMs in each list have the same value of selected criteria).
- Make a descending order of the lists according to selected criteria values.
- Browse all lists of higher criteria value to the smallest one (As long as there is a list).
- For each list:
  - ❑ If another criteria is selected:
    - ✓ Execute next activity.
  - ❑ Otherwise, If the current criteria corresponds to the last iteration:
    - ✓ For each list, browse all its preventives maintenances (PMs)
      - o for each PM in a list:
        1. If the PM is flexible: Start at the beginning of the period of work balancing.
        2. If the PM is not flexible : Start at the scheduled time.
        3. Check the appropriate availability condition:

           (Craft-men hours for each PM Craft) + (Craft-men hours total already balanced for this period "Week") <= (total of craft-men hours for this period) - (total of hours reserved for non-preventive work).
        4. For this PM, do the same verification for all next due period basing on PM frequency.
        5. Check availability:

- If (it is possible to place the PM in all periods relating to this frequency)
  - Mark as PM and placed,
  - Add hours amount of each craft to already balanced charge.
- Otherwise, Skip to the next period (week).

5 Conclusion

Extensibility is one of the most important characteristics and advantages of our approach. The user of this approach can integrate any quantitative or qualitative criteria to have more restricted workload balancing results. Also, we have defined many other extension points. Especially, the condition that verify the craft-men availability. User can define the appropriate condition for its business process. A formal approach allows implementing performance and quality metrics to evaluate and enhance resulted scenarios. The purpose of the application of the measures in preventive maintenance is to ensure that everything is under control. However, they should also be used to highlight the problems and irregularities to obtain desired behaviors and areas for improvement.

The right scenario will be selected and approved for maintenance workload balancing. If changes are happened on production constraints, appropriate adjustments could be applied on control dada to have the right balancing. When the labor supply exceeds the workload, everything is under control. When the workload is more than the availability of craft-men, it will be necessary to reduce some non-essential at this time, or increase the availability of resources activities.

This approach provides a strong plate-form for Computerized Maintenance Management Systems (CMMS) to enhance the work balancing task in the maintenance management process.

References

1. Cassady, C.R., Kutanoglu, E.: Integrating Preventive Maintenance Planning and Production Scheduling for a Single Machine. IEEE Transactions on Reliability 54(2) (2005)
2. Ross, D.T.: Structured Analysis (SA): A Language for Communicating Ideas. IEEE Transactions on Software Engineering SE-3(1) (January 1977)
3. Shapiro, J.F.: Mathematical programming models and methods for production planning and scheduling. In: Graves, S.C., Rinnooy Kan, A.H.G., Zipkin, P.H. (eds.) Handbooks in Operations Research and Management Science. Logistics of Production and Inventory, vol. 4. North-Holland (1993)
4. Valdez-Flores, C.: Survey of preventive maintenance models for stochastically deteriorating single-unit systems. Nav. Res. Logist. 36(4), 419–446 (1989)
5. Ashayeri, J., Teelen, A., Selen, W.: A production and maintenance planning model for the process industry. Int. J. Production Res. 34(12), 3311–3326 (1996)
6. Sortrakul, N., Nachtmann, H.L., Richard Cassady, C.: Genetic algorithms for integrated preventive maintenance planning and production scheduling for a single machine. Computers in Industry 56(2) (2005)

Computer Support for the Railway Safety Management System – Requirements Analysis

Andrzej Białas

Institute of Innovative Technologies EMAG, 40-189 Katowice, Leopolda 31, Poland,
a.bialas@emag.pl

Abstract. The chapter concerns the identification of requirements to build software supporting the Safety Management Systems (SMSes) used by railway undertakings or infrastructure managers. The discussed SMS should be in conformance to the European regulations. All relevant regulations defining the SMS are reviewed, their elements requiring computer support are identified and analysed whether they can be implemented on the OSCAD software platform. This analysis is needed because OSCAD was elaborated originally as a software tool supporting other management systems, i.e. business continuity and information security management systems. The set of requirements and their implementation methods are elaborated. The most favourable functions requiring software support are: risk management, incident management, safety indicators, documents and tasks management and reporting. This work is an input to build the experimental system and to conduct its validation.

Keywords: railway transportation, safety management system, business continuity, risk management, incident management, computer support for management systems.

1 Introduction

The chapter concerns computer support of a safety management system (SMS) designed for the railway transportation. The aim of the chapter is to analyse the European regulations and to identify the requirements for the SMS computerisation.

The basic laws related to the railway transportation safety are included in the directive [1] and in other European regulations, like [2] (risk management), [3] (safety monitoring), [4] (infrastructure manager – conformity), [5] (safety indicators). All these regulations will be analysed to answer the following questions:

- Which areas, processes or actions of the SMS are suitable for computerization, i.e. for replacement or considerable support of human labour by software?
- What kind of advantages (cost, time, safety, quality, etc.) can it bring?

The directive on the safety of the Community's railways [1] applies to railway systems in the member states. According to this directive, "the safety management system means the organisation and arrangements established by an infrastructure

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_7,

manager or a railway undertaking to ensure the safe management of its operations". The railway undertaking is understood as a subject who provides transport of goods and/or passengers. The [1] covers safety requirements on the system as a whole, including safe management of infrastructure, traffic operations and interaction between railway undertakings and infrastructure managers. It is supplemented by other regulations. The directive introduces the key terms for SMSes:

- Common safety targets (CSTs), i.e. "safety levels that must at least be reached by different parts of the rail system (such as the conventional rail system, the high speed rail system, long railway tunnels or lines solely used for freight transport) and by the system as a whole, expressed in risk acceptance criteria" – they play the role of safety requirements for the railway transport system,
- Common safety methods (CSMs) – describe "how safety levels and achievement of safety targets and compliance with other safety requirements are assessed", the CSMs concern different aspects of the SMS: risk management, monitoring, etc.
- Common safety indicators (CSIs), characterising accidents, incidents, technical and management issues, and used to measure safety performance and to facilitate the economic impact assessment of CSTs.

Particular railway subsystems should co-operate according to technical specifications for interoperability (TSIs). There are specific technical regulations concerning: infrastructure, traffic management, energy, rolling stock, maintenance, control command and signalling, telematics, noise, etc.

Infrastructure managers and railway undertakings are responsible for the safety of their own parts of the entire safety system. It implies strong cooperation between these two groups. They ought to establish their own SMSes to assure safety of the railway transport of passengers and goods.

The objective of the [2] is to guide the risk management process with respect to the SMSes, especially:

- To agree on common safety methods (CSMs) covering at least risk evaluation and assessment methods (based on the European Railway Agency recommendations),
- To support railway companies in their efforts to establish safety management systems in order to ensure that the railway system can achieve at least the CSTs.

The [3] is focused on the common safety method (CSM) for the SMS monitoring during its operation. The [4] defines the common safety method (CSM) for assessing conformity with requirements to get the safety authorisation for the system, while the directive [5] specifies common safety indicators (replacing and extending their older specification placed in the [1]). There are more laws about railway transportation, including national ones [6], [7].

The specification of the SMS (document-based) of Polish railway undertakings is presented in [8]. The process approach is preferred and, apart from the main transportation process, other supported processes are identified.

The paper [9] presents the safety approach promoted by the Swiss Federal Office of Transport, exemplified by 3 cases (risk-oriented placement of train protection

systems, level crossing protection, wayside train monitoring systems). The quantitative risk analyser is able to consider the cost of safety measures.

The paper [10] features a quantitative risk analysis method used in East Japan Railway Co. separately to stations and level crossings. The method, ISO 31000 compatible, considers 3 phases: risk exposures, incidents and accidents, and the probability to pass to the next phase. The tool is developed in their own research laboratory.

The paper [11] presents Canadian experiences in the SMS implementation according to the Canadian Railway Safety Act (RSA). This SMS has similar components to the European one. The education, communication and safety culture of the involved personnel are the key issues at the beginning, before the risk management, performance measures, monitoring, etc. are implemented. The works are focused on regulations and their implementation. No specialised supporting tools have been used up until now. One of the conclusions from the review of the RSA implementation points out the elaboration of the user-friendly SMS tools for small railway companies.

The paper [12] discusses the development process of the software for safety-related signalling system implemented in Beijing National Railway Design and Research Institute of Signal and Communication. The proper software life cycle management allows to meet high RAMS (reliability, availability, maintainability, safety) performance requirements and conformity to the European EN 50128 standard. The special UML based software framework was elaborated allowing for: the requirements management, development assistance for safety-related signalling software, automatic testing, knowledge base support, etc.

The paper [13], based on experiences in Hong-Kong, China, UK, and Australia, presents how to integrate the Human Factors Management (HFM)- and SMS processes, i.e. to make the SMS more fit for people. Human factors (HF) are considered during the design – in every safety barrier, which helps to prevent incidents. Human related hazards are systematically identified and mitigated by relevant safety and protection measures. HF studies were integrated with the incident review by using a special tool – all incidents and undesirable events are registered in the Operational Data Management System, which considers HF taxonomy.

The SMS is usually implemented as a documentation-based organisational and procedural system. There are no specialised software tools supporting management processes, except risk management. The safety managers have to perform many laborious activities. Most of these activities are repeatable and related data resources are potentially reusable – their computerisation can bring many advantages. Safety managers have to perform rather complicated analyses for which specialised tools are needed. Additionally, they have to master huge and diversified documentation which is usually scattered throughout the entire railway company and its partners – it needs right management. Safety managers have to gather and analyse a huge amount of data, a part of which is related to safety measures. All these factors are strong arguments for the automation of safety management systems in railway transportation.

The paper is aimed at the identification of activities which can be supported by the software to reduce costs and time. An improved management system means increased safety of the rail transport system.

Section 2 presents the safety management system implied by the European regulations and areas possible to computerise. The main section identifies the SMS activities and shows how to implement them on the OSCAD software platform. It leads to the identification of the OSCAD-SMS requirements. In the conclusions the works on the OSCAD-SMS requirements identification are summarised. On this basis the prototype of the OSCAD-SMS has been developed and validated. Finally, the validation plan and results are summarised – they will be discussed in detail in a separate publication.

2 Identification of the Safety Management System (SMS) Areas Possible to Computerise

The detailed analysis of European laws with respect to the SMS computerisation brings the following general conclusions (general requirements):

1. To ensure legal conformity, a general, software supported, SMS management framework should be developed, encompassing the SMS elements specified in the [1]/Annex III.
2. To achieve authorisation and accountability, this framework should include role based administration facilities.
3. To master different kinds of SMS documentation, this framework should include documentation and reporting facilities.
4. To ensure evidences allowing to assess the SMS conformity to the law, the framework should be able to store and manage records of operations.
5. To manage activities of the involved persons, the task management, scheduling and communicating facilities should be helpful.
6. To properly monitor the SMS operations, the audit and review facilities should be built in to the framework, according to the [3]/Annex.
7. To monitor the infrastructure manager SMS conformity to the legal regulations [4], a specific self-assessment facility can be added to the main framework (an optional solution).
8. To properly manage risk, the risk management subsystem should be built in to the framework, according to the [2]/Annex I.
9. To monitor the SMS effectiveness and the rail transport safety, the framework should be equipped with the facility to gather, analyse and present the information used as the safety indicators, according to the [5] Annex I.

These requirements are very general and each concerns one dedicated part of the complex IT system. A more detailed analysis is needed for each of them. To develop the computer supported SMS, the top-down development methodology is applied.

3 Feasibility Analysis for the Identified Requirements

The computer supported safety management system can be developed as a dedicated system compliant with the users' requirements or can be implemented on the existing

software platform. Here, the second approach is considered, and the OSCAD software [14] is used as the implementation platform.

OSCAD was developed at the Institute of Innovative Technologies EMAG within a project co-financed by the National Centre for Research and Development (NCBiR). The project acronym means "Open, scalable, and integrated, computer aided system for business continuity and information security management". The OSCAD software system manages business continuity according to BS25999 (ISO 22301) and information security according to ISO/IEC 27001. OSCAD is able to:

- Monitor the factors which cause crisis situations in institutions, i.e. when there are disturbances in the continuity of business processes or breaches in information security due to threats which exploit certain vulnerabilities,
- Reduce negative impact (consequences) of business continuity disturbances or information security breaches,
- Support the recovery of a business process to its original form after incidents.

OSCAD is focused on the protection of business processes and information assets. Due to its openness and configurability it can be used in different application domains to protect other kinds of processes and assets, e.g. people and infrastructure protection against flood [15], [16], coal extraction process and engaged resources [17]. Successful experimentations in these domains were a motive to experiment in other domains too. Thus co-operation began with a big railway undertaking to initiate a case study on the SMS implementation on the OSCAD platform.

These experimentations should answer the following questions:

- Is it possible to implement the SMS on the OSCAD platform and to develop a satisfactory solution for the railway companies (OSCAD-SMS)?
- What range of the OSCAD software modification/extension is needed?

If experiments give positive results, the road map of the OSCAD-SMS implementation should be specified, if not –precise conclusions will be provided how to build dedicated software for SMS based on the users' requirements.

General requirements related to the SMS computerisation and implied by the laws (Section 2) are grouped in 3 sets, which ought to be refined and evaluated against the OSCAD features and functions [14].

3.1 SMS Management Framework

The first set encompasses the requirements 1-7 and concerns the general SMS management framework.

Annex III of the directive [1] lists the basic SMS elements. This is a specification of the documents based (paper or electronic) management system. All managing processes, their input and output, are specified as different kinds of documents. The office software can be used as computer supporting tools. Recently, such systems have been elaborated and now are used and refined in many European railway companies.

The basic elements of the SMS [1]/Annex III are shown in Table 1. They create a management framework and are numbered (F.1-F.15). They can be considered elements of the document based system. The computerisation of the SMS with the use of the OSCAD platform required justification – what kind of an added value it can bring for railway companies (increasing quality and safety, decreasing management efforts, time, cost).

The basic added value is the ability to manage numerous documents. Safety policy, procedures, instructions, targets, plans, standards, TSIs, authority decisions, etc. can be attached to OSCAD in the right places, i.e. where access to them is needed. Every document has its own descriptor (identifier, version, date of.., owner, approved by.., history, etc.) for proper management (versioning, approval, right dissemination, assigning to activity, formatting, etc.). Yet, the documents management ability seems to be an insufficient argument to implement OSCAD or similar software. For this reason in the last column of Table 1 there are additional OSCAD features allowing to: better manage laborious activities, increase management-related data quality, gain data reusability effect, support complex operations, provide better communication, etc.

Table 1. Implementation of the basic SMS elements in the management framework

No.	Requirement ([1]/Annex III)	Added value due to the OSCAD implementation
F.1	A safety policy; it should be approved by the organisation's chief executive and communicated to all staff	Documents are attached to the system, managed and distributed, process modelling, asset inventory, roles, permissions, access control, communication facilities, dictionaries
F.2	Qualitative and quantitative targets of the organisation for the maintenance and enhancement of safety, and plans and procedures for reaching these targets	Event/incident[1] management, statistics, processes management, measures and indicators, task management, planning
F.3	Procedures to meet existing, new and altered technical and operational standards or other prescriptive conditions as laid down in TSIs, national or other relevant safety rules or authority decisions	Review and audit facility, checklists
F.4	Procedures to assure compliance with the standards and other prescriptive conditions throughout the lifecycle of equipment and operations	Review and audit facility, checklists, scheduling
F.5	Procedures and methods for carrying out risk evaluation and implementing risk control measures whenever a change of the operating conditions or new material imposes new risks on the infrastructure or on operations. To manage changes in equipment, procedures, organisation, staffing or interfaces, the entities in charge of maintenance should have risk assessment procedures.	4 risk analysers (process-oriented impact analyser and cause analyser, asset-oriented impact analyser and cause analyser, configurable risk factors measures, dictionaries, possible operational- and occupational health/security risks analysis

[1] In OSCAD the term "incident" has a more general meaning; it can encompass categories used in the SMSes, like: event, accident, serious accident, incident.

Table 1. *(continued)*

F.6	Provision of programmes for training of staff and systems to ensure that the staff's competence is maintained and tasks carried out accordingly	Training management facility, scheduler, task manager, asset inventory (personnel)
F.7	Arrangements for the provision of sufficient information within the organisation and, where appropriate, between organisations operating on the same infrastructure	Roles, permissions, access control, communication facilities, communication between OSCAD systems of co-operating railway institutions
F.8	Procedures and formats for how safety information is to be documented and designation of procedure for configuration control of vital safety information	Configurable document/report templates, process management
F.9	Procedures to ensure that accidents, incidents, near misses and other dangerous occurrences are reported, investigated and analysed and that necessary preventive measures are taken	Incident management subsystem, incident statistics, dictionaries, asset inventory, process management, measures and indicators, support for the corrections and improvements
F.10	Provision of plans for action and alerts and information in case of emergency, agreed upon with the appropriate public authorities	Emergency planning (based on the BCP – business continuity planning), communication interfaces
F.11	Provisions for recurrent internal auditing of the safety management system	Review and audit, checklists, scheduling, self-assessment of the conformity
F.12	Documented responsibilities within the railway organisation	Roles, asset inventory (personnel is one of the asset categories)
F.13	Documented role of the management and the staff involvement	Roles, specification of the SMS owner institution in the system
F.14	Documented demonstration of the continuous improvement of the SMS	Reports from the continual improvement process which is based on the measures and indicators
F.15	The introduction of measures to encourage improvements in the safety and health of workers at work and its relevant individual directives are fully applicable to the protection of the health and safety of workers engaged in railway transport	Risk analyser – analyses related to the occupational health and safety (OHSAS)

Apart from the document management and abilities listed in Table 1, the OSCAD platform offers other facilities, which can be considered common added values:

- The reporting subsystem – some activities and their results can be reported on the configurable templates basis; reports can be issued for the national safety authority;
- The records subsystem – all relevant activities and their results are registered by the system to generate evidences for the certification or authorisation processes,
- The task manager (a simple workflow) – a task can be issued by actors (users), by the measure and indicator facilities (on alert) or by the time scheduler.

The above analysis presents an ability of the OSCAD system to perform different SMS tasks implied by the requirements placed in the European regulations.

3.2 Risk Management Subsystem

The second set includes only one requirement from Section 2 (general requirement No. 8). It concerns the risk management and can be considered the refinement of the F.5 and F.15/Table 1 according to the [2]/Annex I. Refined requirements and their implementations are presented in Table 2.

Table 2. Implementation of the basic SMS elements related to the risk management

No.	Requirement ([2]/Annex I)	Added value due to the OSCAD implementation
R.1	Change management allowing to assess the range and significance of changes	Procedural solution – not supported in the OSCAD software, but possible
R.2	Risk management system definition and configuration	Specification of a railway institution, roles, interface, dictionaries, risk scales, formulas, acceptance criteria, etc.
R.3	Risk acceptance based on the application of codes of practice	Possibility to define most preferable measures (best practices) for a given hazard. Statement of Applicability (SoA) mechanism (drawn from ISO/IEC 27001) can be used.
R.4	Risk acceptance based on the comparison with similar parts of the railway system	The risk analyser can define reference institutions with risk and safety measures implemented.
R.5	Risk acceptance based on 'explicit risk estimation' principle. It is frequently used for complex or innovative changes.	The basic mode of operation is explicit risk estimation and selecting risk-adequate safety measures. Up to 5 variants of safety measures can be considered and 1 is selected as the implementation target .
R.6	Different focus of the risk analysis: operational risk and for occupational health safety systems (OHSAS) risks analysis, third party	Process-/Asset oriented impacts analyser (qualitative) and causes analyser (quantitative). Configurable risk factors scales, models, dictionaries, threats, vulnerabilities, safety measures including preferable ones (best practices). Possible analyses for the OHSAS
R.7	Hazard records	In OSCAD these are risk scenarios. Hazard taxonomy is compliant with event/incident taxonomy
R.8	Risk control management and audits	Risk treatment plans review, audit, measures and indicators
R.9	The exchange of safety-relevant information between different actors within the rail sector in order to manage safety across the different interfaces which may exist within this sector	Reporting, communication facilities, incident statistics
R.10	The evidence resulting from the application of a risk management process	Recording, reporting, archiving, software built in the acceptance procedure

The harmonisation of the European rail transport system requires the unified methodology for risk evaluation and assessment expressed by the CSM [2]. The railway undertakings and infrastructure managers should work in compliance with different technical standards, e.g. EN 50129.

3.3 Safety Indicators Subsystem

The third set includes only one requirement from Section 2 (general requirement No. 9). It concerns the safety indicators and can be considered the refinement of the F.9 and F.14 /Table 1 according to the [5]/Annex I.

Common safety indicators (CSIs) allow to assess the safety performance of the SMS and the economic impact of CST. The CSIs are expressed as total-/relative to train-kilometres values. The following main categories of indicators are distinguished:

1. Indicators related to accidents.

1.1. Number of significant accidents break-down into types/subtypes (collisions, derailments, accidents to persons, fires in rolling stock, etc.).

1.2. Number of persons seriously injured and killed by types of accidents divided into the following categories: passengers, employees, contractors, level crossing users, unauthorised persons on railway premises, others.

2. Number of accidents involving the transport of dangerous goods.

3. Number of suicides.

4. Indicators related to precursors of accidents, such as the number of: broken rails, track buckles, signalling failures, signals passed at danger, and broken wheels and axles on the rolling stock in service.

5. Indicators to calculate the economic impact of accidents: number of deaths and serious injuries, cost of damages to the environment, cost of material damages to the rolling stock or infrastructure, cost of delays as a consequence of accidents.

6. Indicators related to technical safety of infrastructure and their implementation.

6.1. Percentage of tracks with Automatic Train Protection (ATP) in operation, percentage of train-kilometres using operational ATP systems.

6.2. Number of level crossings by types of protection.

7. Indicators related to safety management – internal audits performed by infrastructure managers and railway undertakings.

The OSCAD added value concerns sampling and storing relevant data, analysing them and reporting. CSIs are implemented in OSCAD using its measures and indicators facility. For each category/subcategory the variables are defined, which sample relevant values. The given measure value can be entered manually by the user, calculated, or delivered automatically, e.g. from telematics applications. Most data are sampled by the incident management and statistics facility. During the SMS operation the information records related to events/incidents are sampled on-line. Special categorisation of terms for railway transport is elaborated. This categorisation is based on the common taxonomy of the risk record (prognosis) and incidents (reality).

Gathering and publishing information on common safety indicators (CSIs) allow to monitor the railway safety progress and to assess the degree of the CSTs achievements. Each year a special safety report should be submitted to the railway authority. Railway undertakings must hold safety certificates in order to access the railway infrastructure. Infrastructure managers must obtain safety authorisations to manage and operate the rail infrastructure. The rolling stock should be properly authorised, while the train drivers and staff accompanying the trains should be sufficiently trained.

4 Conclusions

The chapter presents the first step of works on the computer supported safety management system (SMS) for rail transportation. The ready-to-use OSCAD software has been chosen as an implementation platform. Straight SMS implementation was not obvious as OSCAD was developed for such application domains as business continuity- and information security management. To resolve this issue, a requirements analysis (presented in this chapter) and validation on near real data were performed.

The chapter reviews the European regulations for rail transport safety. On this basis SMS requirements relevant to their computerization were identified and their implementation on the OSCAD platform discussed. To achieve better management quality, increased safety, decreased time and cost of operations, the works are focused on:

- The complicated operations, analyses where reusability of data is easily gained,
- The laborious, complex management activities, when mistakes are possible.

The next step is the validation of OSCAD-SMS in co-operation with a railway undertaking, including: analysing the existing SMS documents, specification of the organization in OSCAD, roles, dictionaries, risk analysers setup, processes, assets, performing different risk analyses and safety measures selection, event/incident import, creating reports and statistics, etc. This huge work to perform will be discussed in a separate publication [18], while in this paper only general conclusions are placed:

- The results of validation are generally positive but the complete SMS has not been finalised yet – only examples of important data were introduced/generated and only the key functions validated;
- The OSCAD platform was flexible enough to implement SMS requirements but menu functions and user messages were adapted to the terminology used in the rail transport safety domain; besides, databases were extended because more information is needed to specify incidents and related assets.

The personnel still play the key role in SMS. They cannot be eliminated in the management processes, but can be considerably supported. The SMS computerisation is possible only when the traditional, document based SMS achieves its maturity.

References

1. Directive 2004/49/EC of the European Parliament and of the Council of 29 April 2004 on safety on the Community's railways and amending Council Directive 95/18/EC on the licensing of railway undertakings and Directive 2001/14/EC on the allocation of railway infrastructure capacity and the levying of charges for the use of railway infrastructure and safety certification (Railway Safety Directive)
2. Commission Implementing Regulation (EU) No 402/2013 of 30 April 2013 on the common safety method for risk evaluation and assessment and repealing Regulation (EC) No 352/2009

3. Commission Regulation (EU) No 1078/2012 of 16 November 2012 on a common safety method for monitoring to be applied by railway undertakings, infrastructure managers after receiving a safety certificate or safety authorisation and by entities in charge of maintenance
4. Commission Regulation (EU) No 1169/2010 of 10 December 2010 on a common safety method for assessing conformity with the requirements for obtaining a railway safety authorisation
5. Commission Directive 2009/149/EC of 27 November 2009 amending Directive 2004/49/EC of the European Parliament and of the Council as regards Common Safety Indicators and common methods to calculate accident costs
6. European Railway Agency, http://www.era.europa.eu/Pages/Home.aspx
7. Office of Rail Transportation (Poland), http://www.utk.gov.pl/en
8. Sitarz, M., Chruzik, K., Wachnik, R.: System zarządzania bezpieczeństwem polskich operatorów kolejowych (Integrated Safety Management System of Polish railway undertaking). Czasopismo Techniczne Mechanika, 7-M/2012, Zeszyt 14 rok 109 (January 04, 2014), http://suw.biblos.pk.edu.pl/resources/i1/i5/i1/i0/i4/r15104/SitarzM_SystemZarzadzania.pdf
9. Zeilstra, P.: Moy Ch.: Our Approach to the Safety vs. Cost Issue. In: 22nd International Railway Safety Conference (IRSC), London (2012)
10. Kusukami, K.: Risk Assessment for East Japan Railway Company. In: 21st International Rail Safety Conference (IRSC), Melbourne (2011)
11. Bourdon, L.: Implementing Safety Management Systems in the Canadian rail industry. In: 21st International Rail Safety Conference (IRSC), Melbourne (2011)
12. Chen, L.: Software lifecycle support and management system for safety-related signalling system. In: 22nd International Railway Safety Conference (IRSC), London (2012)
13. Nelson, N.: Making Safety Management Systems fit for humans. In: 22nd International Railway Safety Conference (IRSC), London (2012)
14. OSCAD, http://oscad.eu/index.php/en/knowledge-base/about-management-systems
15. Andrzej, B.: Risk assessment aspects in mastering the value function of security measures. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 25–39. Springer, Heidelberg (2013), http://link.springer.com/chapter/10.1007%2F978-3-319-00945-2_3#page-1
16. Bagiński, J.: Software support of the risk reduction assessment in the valuesec project flood use case. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 11–24. Springer, Heidelberg (2013), http://link.springer.com/chapter/10.1007%2F978-3-319-00945-2_2#page-1
17. Białas, A.: Zarządzanie ciągłością działania oraz bezpieczeństwem informacji i innych zasobów w górnictwie, Mechanizacja i Automatyzacja Górnictwa, Czasopismo Naukowo – Techniczne, Nr 8(510), Instytut Technik Innowacyjnych EMAG, Katowice, pp. 5–64 (English version: pp. 125–138, Russian version: pp. 198–213) (2013)
18. Bialas, A.: Computer support for the railway safety management system – first validation results. In: 9th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX 2014, The Brunow Palace, Poland, June 30-July 4 (accepted, 2014)

Computer Support for the Railway Safety Management System – First Validation Results

Andrzej Białas

Institute of Innovative Technologies EMAG, 40-189 Katowice, Leopolda 31, Poland,
a.bialas@emag.pl

Abstract. The chapter concerns the validation of the railway undertaking Safety Management System (SMS), compliant with the European laws and implemented on the OSCAD software platform. The system is called OSCAD-SMS. The chapter is a continuation of the publication [1] devoted to the system requirements elaboration. OSCAD was elaborated originally as a software tool supporting business continuity and information security management systems. The chapter presents a validation plan encompassing the OSCAD-SMS setup (to prepare the system for operation) and the validation scenario. During the setup the main dictionaries (threats, vulnerabilities, safety measures, roles, etc.), risk parameters, measurement units, and many other elements were defined. The scenario takes into consideration the most important railway SMS operations, like: process management, asset inventory, risk management, incident management, audit, safety indicators, and reporting. The validation confirms the possibility of implementing SMS on this platform to create a computer supported SMS.

Keywords: railway transportation, safety management system, risk management, incident management, computer support for management systems, software validation.

1 Introduction

The chapter deals with computer support of the safety management system (SMS) used in railway transportation. It is a continuation of the publication [1]. According to the European laws, railway undertakings and infrastructure managers should have SMSes deployed.

The requirements for this system were identified by reviewing the essential European laws [1], such as: the directive [2], [3] (risk management), [4] (safety monitoring), [5] (infrastructure manager – conformity), the [6] (safety indicators).

The identified requirements are experimentally implemented on the OSCAD platform [7]. OSCAD – "Open, scalable, and integrated, computer aided system for business continuity and information security management" was developed at the Institute of Innovative Technologies EMAG within a project co-financed by the National Centre for Research and Development (NCBiR). The OSCAD software was originally developed to support business continuity management according to BS25999 (ISO

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_8,

22301) and information security management according to ISO/IEC 27001. It is used to control factors which disturb business processes or breach information assets in an institution (business, public) leading to negative consequences, to limit losses when an incident occurs, and to help in the recovery process.

The solution is open and flexible and thus possible to implement in other application domains, but each time it requires a preliminary study and adaptation. It was confirmed during some experiments performed in the flood protection [8], [9] and coal mining [10] domains. The identified SMS requirements were evaluated against the features of the OSCAD software. It allows to create the experimental OSCAD-SMS system, containing the key SMS functionality and data. Thanks to co-operation with a railway undertaking, the used data are near reality, although are anonymised. The aim of the chapter is to present the validation of the experimental computer supported SMS, called OSCAD-SMS.

As this chapter is a continuation of the publication [1], the state of the art, legal regulations, terminology, and requirements are not discussed here again. The decision to work on the computer support for the railway SMS has been implied by the following situation. The SMSes in Europe (and similarly in other countries) have the organisational and procedural character and are based on paper or electronic documents. The office software, or proprietary risk analysers, are often used as tools. The laws are changed and extended. Therefore, the SMSes are constantly refined, and the people involved are educated to raise the safety culture. Waiting for matured solutions, it is a good time to elaborate the added value – the software support for most complex, laborious and difficult SMS operations.

OSCAD-SMS is focused on the protection of the transport process and the staff. The validation should answer the following questions:

- Is it possible to build SMS on the basis of the OSCAD software?
- Are the requirements [1] adequately selected? Do they bring advantages with respect to the quality, time and cost of the operation, can this lead to better safety of the railway company?

The next section presents the validation plan. The basic use case will be discussed. The OSCAD-SMS setup, preparing the software for operation, will be the first step of the validation (Section 3). Section 4 will present the basic operations on the SMS, such as assets and processes management, risk analysis, incident management, safety indicators. Finally, the validation will be summarised to get answers to the key questions, and to define directions for the further development.

2 Validation Plan

Originally, the validation started on the Polish language version of OSCAD. The used data were in Polish, close to real, but anonymised. To present the validation process in the publication, OSCAD was switched to the English version, but the data were left in Polish. During the validation the OSCAD software was modified – functions and

messages were changed to better express the railway SMS terminology. Besides, the database was extended to better describe incidents and assets.

The validation plan considers the following issues:

1. Preparing the OSCAD instance on the Tomcat server.
2. Analysing the European and national safety authority regulations, railway company documents describing the existing SMS, to extract the relevant data.
3. The OSCAD-SMS setup.
4. Modelling the main transport process and examples of related processes.
5. Introducing or editing examples of key assets of different types.
6. Risk management – different modes of operations.
7. Incident management and statistics.
8. Emergency plans.
9. Audit and reviews.
10. Safety indicators.

The OSCAD software (ver. 1 Revision 3079:20130514) was installed on the Tomcat Apache server. Mozilla Firefox 26.0 web browser was used as an interface. They all work on the notebook Dell Latitude® 6520/4GB under Windows 7® OS. In the next sections the key issues will be discussed – OSCAD-SMS setup, basic steps of the validation, and conclusions.

3 OSCAD-SMS Setup

The OSCAD-SMS setup encompasses operations enabling its normal exploitation.

1. Introducing basic information about the railway undertaking.

After defining the system accounts for the key management personnel, the basic information about the SMS owner's organisation can be entered (Fig. 1). On the left the main software menu is shown. On the right panel, tabs can be selected. One of them (shown) allows to enter the basic information on the railway company to the system.

Apart from the above information, the auxiliary items used by OSCAD-SMS are required (Fig. 2), like: keywords for search operations, kinds of documents, tasks and tests for their categorisation and filtering, inputs and outputs of processes, posts of key management, organisational units, third parties – all co-operating external organisations, used time intervals (each day, monthly, yearly,…), kinds of used measurement units (m, sec, train-kilometer, passenger-kilometer, Euro, kilopascal, etc.).

Protected assets encompass posts (e.g. engine driver) used by the asset oriented risk analyser to assess the occupational health risk for different roles. The right panel of Figure 3 presents some lists of the safety indicators according to the [6]/Annex I (mentioned also in [1]). These indicators are one of the most important facilities of OSCAD-SMS.

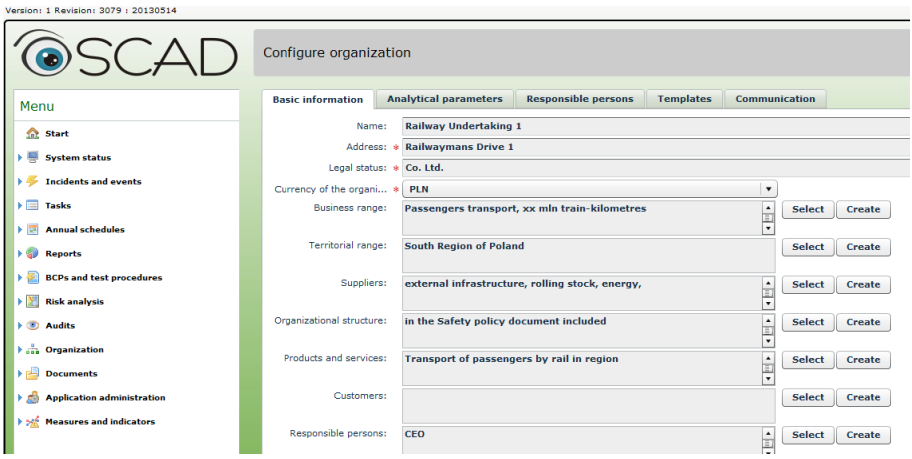


Fig. 1. Basic information about the SMS owner

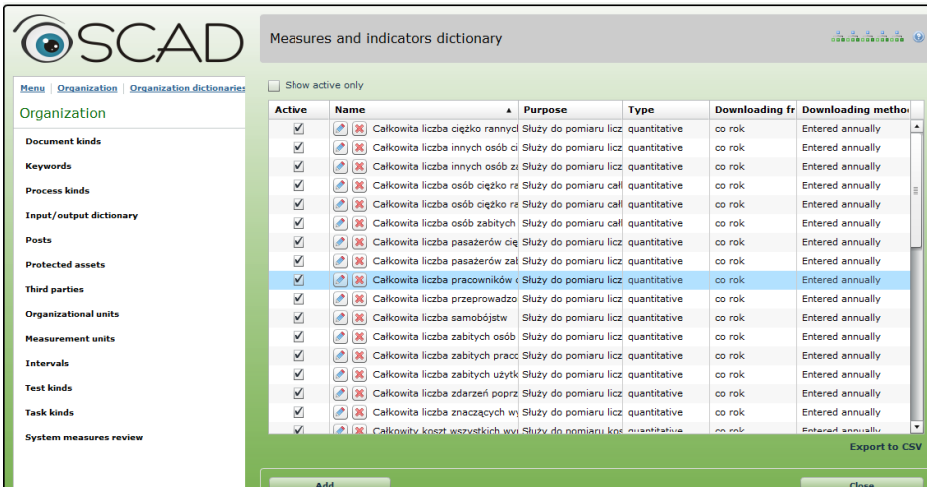


Fig. 2. Auxiliary items used in SMS and safety indicators

2. Predefined lists of threats, vulnerabilities and safety measures.

To perform risk management processes, the system should be equipped with predefined lists of threats and vulnerabilities adequate to the domain of application, though later, during the risk analysis, new items can be still added.

Figure 3 presents a part of the list of threats (in Polish). Each threat item has a short name and longer description. The threat item expresses external factors and related consequences. Similarly, the lists of vulnerabilities are specified. They express weaknesses of the protection system raising the probability of the threat materialization.

Each railway company should elaborate its own list of threats and list of vulnerabilities describing its environment.

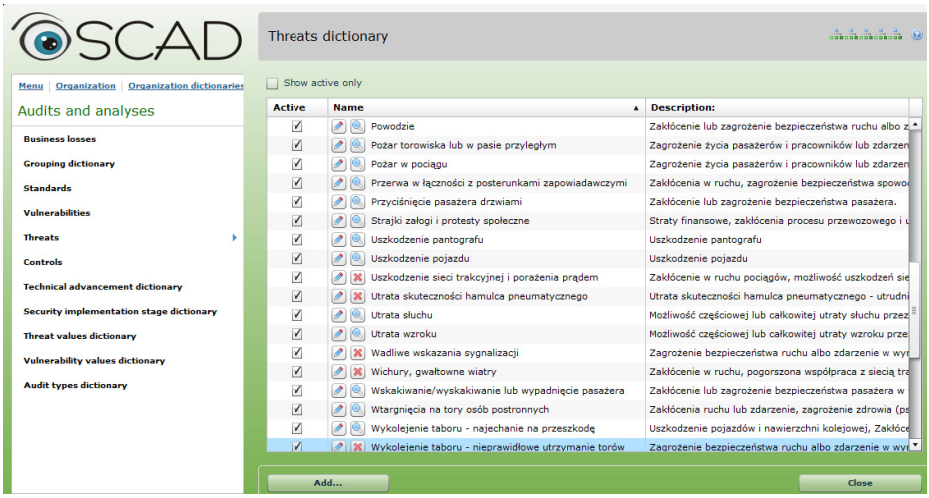


Fig. 3. Predefined threats

The third predefined list contains different kinds of predefined measures (called controls). Some of them concern operational risk (process oriented) and have the RO prefix. Others concern asset oriented occupational risk mitigation (prefix RZ). The prefixes allow to distinguish both groups of measures belonging to different domains of application. This way certain shortcomings of OSCAD are eliminated (inability to manage several lists of measures). Each measure has two cost parameters: yearly depreciation and yearly maintenance cost.

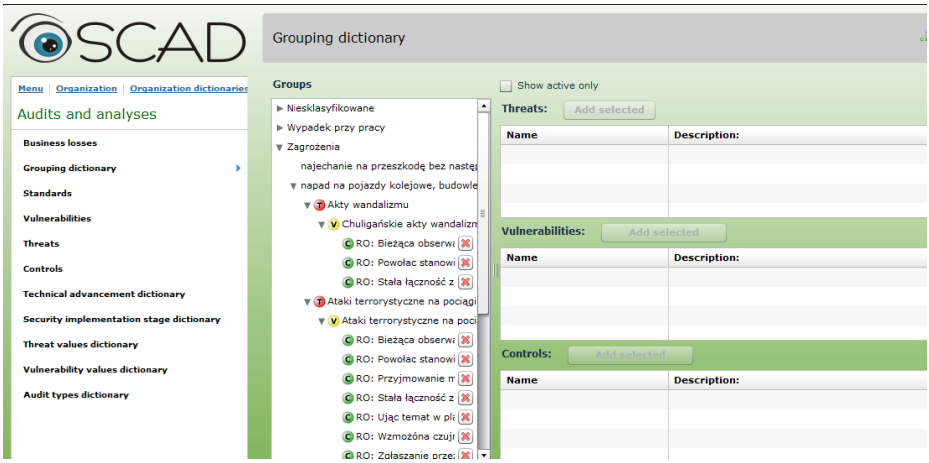


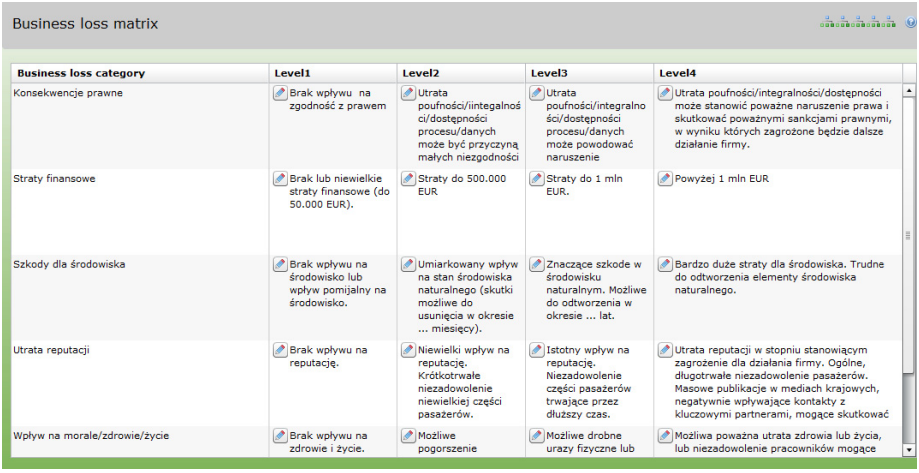
Fig. 4. Grouping dictionary

The grouping dictionary (Fig. 4) allows to assign typical vulnerabilities to a given threat, and safety measures to pairs <threat, vulnerability>. It is very useful during risk management, as it speeds up the selection of relevant safety measures.

3. Configuring the risk manager.

Two kinds of process oriented risk analyses are possible for the processes:

- high level risk analysis (BIA – business impact analysis) to asses consequences,
- low level risk analysis (detailed risk analysis) to assess causes.



Business loss matrix

Business loss category	Level1	Level2	Level3	Level4
Konsekwencje prawne	Brak wpływu na zgodność z prawem	Utrata poufności/iintegalności/dostępności procesu/danych może być przyczyną małych niezgodności	Utrata poufności/integralności/dostępności procesu/danych może powodować naruszenie	Utrata poufności/integralności/dostępności może stanowić poważne naruszenie prawa i skutkować poważnymi sankcjami prawnymi, w wyniku których zagrożone będzie dalsze działanie firmy.
Straty finansowe	Brak lub niewielkie straty finansowe (do 50.000 EUR).	Straty do 500.000 EUR	Straty do 1 mln EUR.	Powyżej 1 mln EUR
Szkody dla środowiska	Brak wpływu na środowisko lub wpływ pomijalny na środowisko.	Umiarkowany wpływ na stan środowiska naturalnego (skutki możliwe do usunięcia w okresie ... miesięcy).	Znaczące szkode w środowisku naturalnym. Możliwe do odtworzenia w okresie ... lat.	Bardzo duże straty dla środowiska. Trudne do odtworzenia elementy środowiska naturalnego.
Utrata reputacji	Brak wpływu na reputację.	Niewielki wpływ na reputację. Krótkotrwałe niezadowolenie niewielkiej części pasażerów.	Istotny wpływ na reputację. Niezadowolenie części pasażerów trwające przez dłuższy czas.	Utrata reputacji w stopniu stanowiącym zagrożenie dla działania firmy. Ogólne, długotrwałe niezadowolenie pasażerów. Masowe publikacje w mediach krajowych, negatywnie wpływające kontakty z kluczowymi partnerami, mogące skutkować
Wpływ na morale/zdrowie/życie	Brak wpływu na zdrowie i życie.	Możliwe pogorszenie	Możliwe drobne urazy fizyczne lub	Możliwa poważna utrata zdrowia lub życia, lub niezadowolenie pracowników mogące

Fig. 5. Business loss matrix

BIA requires the definition of the loss matrix with respect to integrity (I) and availability (A). The loss matrix (Fig. 5) has several loss categories defined (laws consequences, financial, environmental, reputational, human life and health) – placed vertically and the levels to measure losses for them – placed horizontally. Numbers of categories and levels are configurable.

The detailed risk analysis is based on the following formula:

$$Risk\ level = \frac{level\ of\ impacts * probability\ of\ occurrence}{sec.measures\ implementation\ level * sec.measures\ advancement\ factor}$$

Each of four parameters is measured with the use of configured scales predefined for the institution. The level of impacts is assessed for a given threat, but probability of occurrence is assessed for a considered vulnerability – based on the predefined enumerative scales (e.g.: very high, high, low). The security measure implementation level (e.g.: planned, under implementation, fully deployed) and security measure advancement factor (e.g. procedural, technical, fully automated) express the properties of the existing measures during risk scenario assessment.

Besides, the risk acceptance level is assigned as well as many other details.

4 Validation Process

The validation scenario encompasses a few operations important for SMS.

1. Modelling Processes.

SMS is focused on the protection of the transport process. To protect this major process, it is vital to protect other kinds of processes too. In a freight railway company the main transport process is assisted by processes supervising the transport of dangerous materials and special products. In all railway companies the main process is supported by auxiliary processes, responsible for the co-operation with infrastructure managers, transport companies, and suppliers, for the maintenance of rolling stock and technical resources. Internal SMS processes related to the risk analysis, technical and operational risk, occupational risk, and third parties risk are distinguished too, due to their critical meaning. Each process may have subprocesses. During the validation, all relevant processes were defined, including the main one, which has 4 subprocesses (planning, preparedness, performance, completion). Each process can have the RTO (recovery time objective) and MTPD (maximum tolerable period of disruption) parameters assigned. Input and output parameters of the processes are defined. The shortcoming is that the visualisation of the processes (and of the organisation structure) does not work properly yet.

2. Asset Inventory.

Two terms are used: assets (to be protected) and resources (used in a company, either to protect assets or to be protected themselves) – all are placed in the asset inventory. The following asset types are distinguished: technical facilities (e.g. vehicle EN57-1007, Katowice-Warsaw line, Warsaw central station), system application (e.g. ERP, booking, control-signalling), service (e.g. energy supply, vehicle maintenance), immaterial assets (e.g. reputation), human resources (e.g. dispatcher, engine driver, guard, CEO). Assets are related to processes, events and incidents, safety indicators, risk managers, and all management processes.

3. Risk Management.

Figure 6 presents analyses of different status (prepared to start, in progress, verified and archived) expressed by the column of icons in the middle of the panel. To protect the main transport process and related ones, two process oriented analyses are performed: BIA (consequences) and detailed risk analysis (causes) marked “ARO” (operational risk). To protect personal assets in occupational health management, two asset oriented analyses are suitable: BIA for assets and detailed risk analysis (marked “ARZ”, performed for the engine driver post).

Each analysis encompasses many detailed actions. Figure 7 shows details of an analysis for the engine driver from Figure 6 (see ARZ mark). The related threat is “Loss of eyesight”, vulnerability is “Increased eyesight effort”, existing measure/control: “Lighting adjustment”. Threat- and Vulnerability values, … advancement factor and …implementation level are assessed by selecting proper values in list boxes (here: Threat value = middle, Vulnerab. value = probable, and two others= not applicable in this kind of analysis).

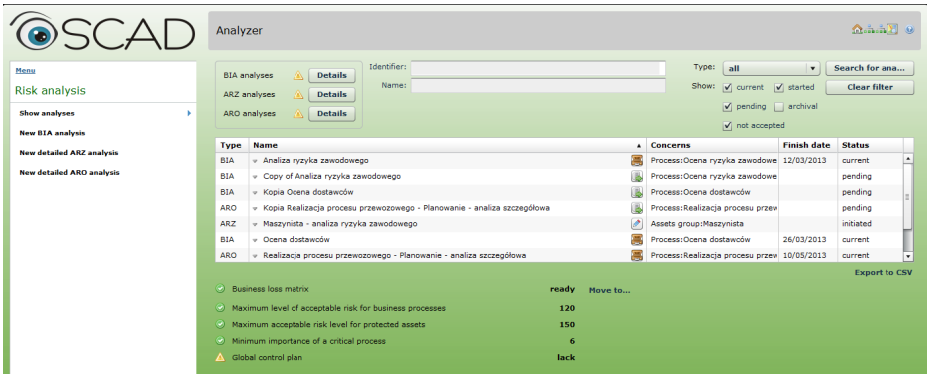


Fig. 6. List of performed analyses

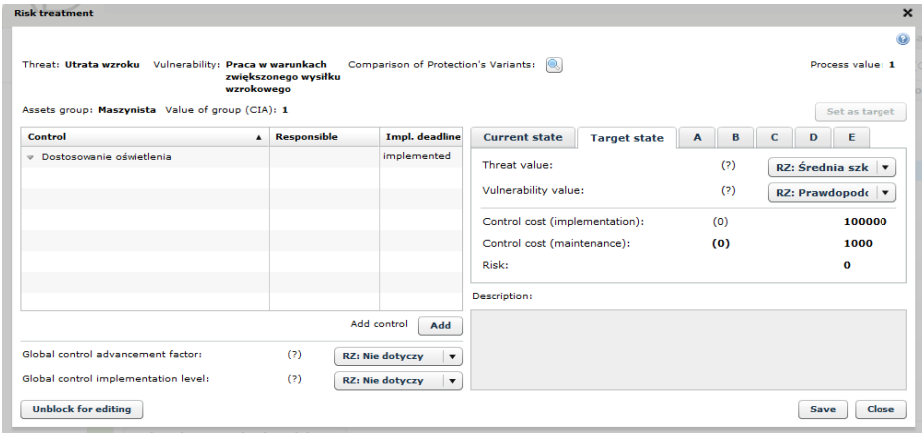


Fig. 7. Occupational risk analysis example

During the risk level assessment the current and the target (expected after the security measures implementation) values of risk are calculated. Up to five variants of security measures can be assessed during the single risk analysis and their risk reduction level and costs can be compared (see Comparison of Protection Variants button), but only one is selected as the target solution.

4. Incident Management, Statistics and Reports.

OSCAD-SMS has a complex event/incident management system. Events are reported, evaluated, classified, while occurrence causes and situation are specified. Based on the reports and statistics, reactive and corrective actions can be planned. Figure 8 shows two simple statistics issued: events by weekdays and kinds of events (police interventions, non-classified, failures, incidents). Event/incident management life cycle was fully implemented, while event/incident database contains relevant data required by laws. Serious incidents activate ready-to-use emergency plans. Of course

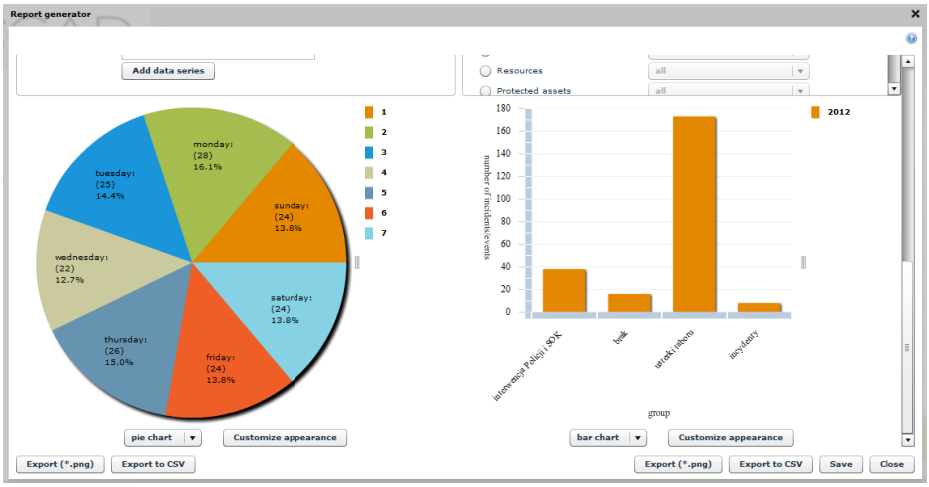


Fig. 8. Events by weekdays and kind of events

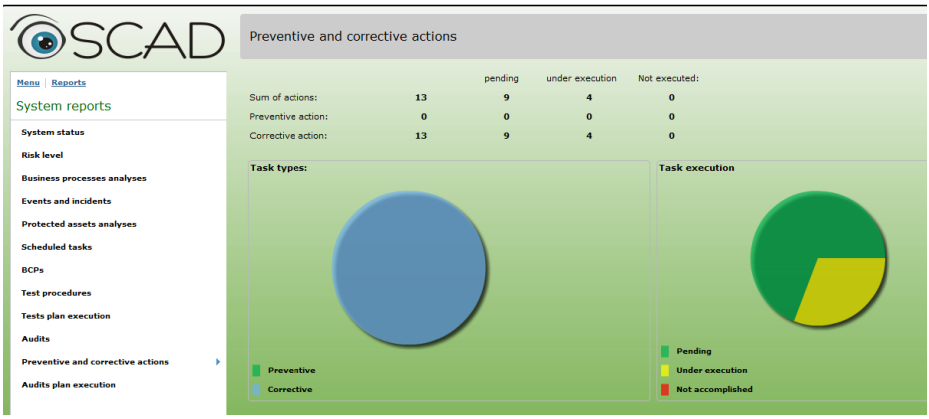


Fig. 9. Report – preventive and corrective actions

these plans are tested and these tests are managed too. Figure 9 shows groups of predefined reports (left side) and a sample report related to corrective/preventive actions. Apart from predefined reports, a configurable report generator has been developed.

5. Audit and Reviews.

OSCAD-SMS supports audit/review planning and management. Figure 10 presents information related to the audit of the main transport process against the regulations elaborated by the infrastructure manager (here: PKP PLK S.A.).

The reported observation points out a corrective action (updating documentation) and the person responsible for it.

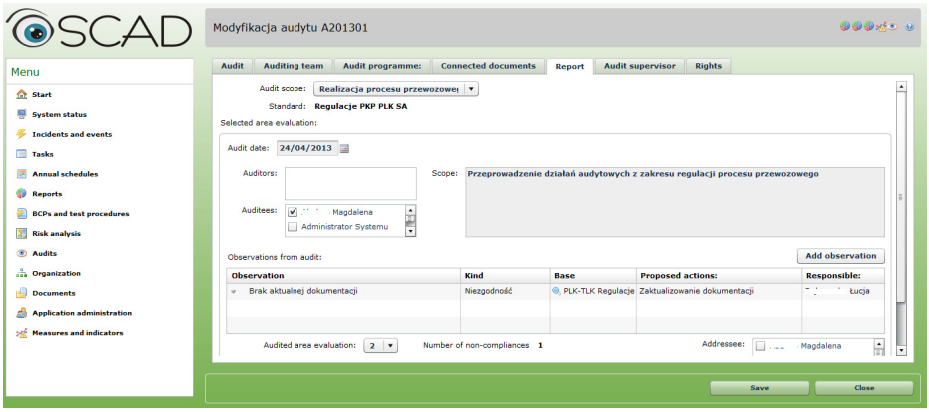


Fig. 10. Audit example

5 Conclusions

The chapter presents the validation process of OSCAD-SMS implementation requirements identified after the analysis of European laws [1], [11].

The validation plan focuses on the key elements of SMS. Scenarios encompass the OSCAD-SMS setup and basic operations. Analysing requirements in Table 1 [1] it is possible to see that the validation deals mainly with: F.1-F.5, F.9, F.11-F.15. For Table 2 [1] the validation includes: R.2, R.5-R.7. Safety indicators [1]/Section3.3 were implemented, but insufficiently exemplified in this chapter.

Contrary to the planned OSCAD-SMS operation in a real environment, the validation process is rather superficial – the range was broad (all key features) but a more detailed analysis is still needed, therefore further experiments should be made. During the validation some conclusions were drawn with respect to the database extension, better reporting, correcting terminology inconsistencies especially in the English version, optimising user efforts during OSCAD-SMS operations, etc.

The validation process gives some observations concerning future deployments. Please note that to deploy the system it is necessary to enter some data into the system. It is quite an effort, yet a single one. Examples of such data are:

- Information about the transportation process, personnel and their training, railway infrastructure and rolling stock, third parties,
- Standards, regulations, forms, instructions, maintenance and system documentation,
- Threats and vulnerabilities identified during the risk analysis.

The deployment will be easier when a traditional, document based, stable system exists in the railway company.

The effort and advantages are similar to the CAD/CAM systems deployments (reusability, quality, decreased time/cost in the time perspective).

References

1. Bialas, A.: Computer support for the railway safety management system – requirements analysis. In: 9th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX 2014, The Brunow Palace, Poland, June 30-July 4 (accepted, 2014)
2. Directive 2004/49/EC of the European Parliament and of the Council of 29 April 2004 on safety on the Community's railways and amending Council Directive 95/18/EC on the licensing of railway undertakings and Directive 2001/14/EC on the allocation of railway infrastructure capacity and the levying of charges for the use of railway infrastructure and safety certification (Railway Safety Directive)
3. Commission Implementing Regulation (EU) No 402/2013 of 30 April 2013 on the common safety method for risk evaluation and assessment and repealing Regulation (EC) No 352/2009
4. Commission Regulation (EU) No 1078/2012 of 16 November 2012 on a common safety method for monitoring to be applied by railway undertakings, infrastructure managers after receiving a safety certificate or safety authorisation and by entities in charge of maintenance
5. Commission Regulation (EU) No 1169/2010 of 10 December 2010 on a common safety method for assessing conformity with the requirements for obtaining a railway safety authorisation
6. Commission Directive 2009/149/EC of 27 November 2009 amending Directive 2004/49/EC of the European Parliament and of the Council as regards Common Safety Indicators and common methods to calculate accident costs
7. OSCAD, http://oscad.eu/index.php/en/knowledge-base/about-management-systems
8. Andrzej, B.: Risk assessment aspects in mastering the value function of security measures. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 25–39. Springer, Heidelberg (2013), http://link.springer.com/chapter/10.1007%2F978-3-319-00945-2_3#page-1
9. Bagiński, J.: Software support of the risk reduction assessment in the valuesec project flood use case. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 11–24. Springer, Heidelberg (2013), http://link.springer.com/chapter/10.1007%2F978-3-319-00945-2_2#page-1
10. Białas, A.: Zarządzanie ciągłością działania oraz bezpieczeństwem informacji i innych zasobów w górnictwie, Mechanizacja i Automatyzacja Górnictwa, Czasopismo Naukowo – Techniczne, Nr 8(510), Instytut Technik Innowacyjnych EMAG, Katowice, pp. 5–64 (English version: pp. 125–138, Russian version: pp. 198–213) (2013)
11. Sitarz, M., Chruzik, K., Wachnik, R.: System zarządzania bezpieczeństwem polskich operatorów kolejowych (Integrated Safety Management System of Polish railway undertaking). Czasopismo Techniczne Mechanika, 7-M/2012, Zeszyt 14 rok 109 (January 04, 2014), http://suw.biblos.pk.edu.pl/resources/i1/i5/i1/i0/i4/r15104/SitarzM_SystemZarzadzania.pdf

Reductions of Operators in Java Mutation Testing

Ilona Bluemke and Karol Kulesza

Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
I.Bluemke@ii.pw.edu.pl

Abstract. The objective of this chapter is to explore the reduction of computational costs of mutation testing of Java programs by selective mutations – omitting mutants generated for a mutation operator. The approaches to reduce the effort in mutation testing are briefly described. The idea of choosing a mutations operator and omitting mutants generated by it is described, next several experiments, conducted in the Eclipse environment using *MuClipse* and *CodePro* plugins, are presented in details. Two especially designed and implemented tools: *Mutants Remover* and *Console Output Analyser* were also used in experiments. Mutation score was used to evaluate the effectiveness of selective mutation testing.

Keywords: mutation testing, cost reduction, Java testing.

1 Introduction

Mutation testing is a fault based software testing technique that was introduced more than forty years ago. The general idea is that the faults used in mutation testing represent the mistakes made by a programmer so they are deliberately introduced into the program to create a set of faulty programs called mutants. Each mutant program is obtained by applying a mutant operator to a location in the original program. To assess the quality of a given set of tests these mutants are executed against the set of input data to see, if the inserted faults can be detected. A very good survey of mutation techniques was written in 2006 by Jia and Harman [1], they also created a repository [2] containing many interesting papers on mutation testing (last updated in 2011). Recently (2012) Bashir and Nadeem published a survey on object mutation [3].

Mutation testing, briefly presented in section 2, is very effective but it can be computationally expensive. Previous research has investigated the effect of reducing the number of mutants by selecting certain operators, sampling mutants at random, or combining them to form new higher order mutants. These approaches are briefly presented in section 3.

The objective of this paper is to examine what is the impact of arbitrary selection of some mutants generated for Java programs, on the mutation score. Conducted by us experiments are presented in section 4 and some conclusions are given in section 5.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_9,

2 Mutation Testing

The mutation testing is a fault based software testing technique that was introduced in 1971 by Richard Lipton (according to [4]). Surveys on mutation techniques were written e.g. by Jia and Harman [1], Bashir and Nadeem [3] and many papers on mutation testing can be found in a repository [2].

The general idea of mutation testing is that the faults represent mistakes made by a programmer, so they are deliberately introduced into the program to create a set of faulty programs called *mutants*. Each mutant program is obtained by applying a mutant operator to a location in the original program. Typical mutation operators include replacing one operator e. g. '+' by another e.g. '-' or replacing one variable by another. To assess the quality of a given set of tests the mutants are executed on a set of input data to see, if the inserted faults can be detected. If the test is able to detect the change (i.e. one of the tests fails), then the mutant is said to be *killed*. The input data for test should cause different program states for the mutant and the original program.

A variety of mutation operators were explored by researchers. Some examples of traditional mutation operators for imperative languages are e.g.: statement deletion, replacement of each arithmetic operation with another one, e.g.: "*" with " /", replacement of each Boolean relation with another one, e.g.: > with >=, == . There are also mutation operators for object-oriented languages, for concurrent constructions, complex objects like containers etc., they are called class-level mutation operators. In [3] a survey on the existing object oriented mutation techniques is presented. Another contribution of this work is a survey of available mutation testing tools.

One of the greatest challenges to the validity of mutation testing is the number of mutants that are semantically equivalent to the original program. *Equivalent mutants* produce the same output as the original program for every possible input. For seven large Java programs, 45% of the mutants not detected by the test suite were shown to be equivalent [5]. Equivalent mutants occur when the mutation can never be exercised, its effect is later cancelled out or it is corroded away by other operations in the program [6]. Determining which mutants are equivalent is a tedious activity, usually not implemented in tools. The impact of equivalent mutants is studied in [7]. Techniques have been devised to identify equivalent mutants using program slicing [8], compiler optimization [9], constraint solving [10] and, more recently, impact assessment [7]. Equivalent mutants are still very difficult to remove completely.

Mutation score, defined as a ratio of the number of killed mutants to the total number of nonequivalent mutants, is used to measure a test set's effectiveness. The total number of nonequivalent mutants results from the difference between the total number of mutants and the number of equivalent mutants which cannot be killed.

Mutation testing of software would be difficult without a reliable, fast and automated tool that generates mutants, runs them against a test suit and reports the results. Among several Java mutation tools there are e.g. Muclipse [11], Judy [12], JavaLanche [7]. The number of generated mutants depends on the number of mutation operators available in a mutation tool e.g. in Muclipse there are 43 mutation operators while in Jumble [13], also Java mutation tool, only 7.

Some research has been conducted to reduce the number of mutants by selecting certain operators, sampling mutants at random, or combining them to form new higher-order mutants. Mutant sampling was proposed by Acree [14] and Budd [15] in 1980.

3 Related Work

The problem of reducing the cost of mutation testing was studied in several papers. In [16] Mathur and Wong proposed two techniques limiting the number of mutants: randomly selected or constrained mutation. In experiments they shown that both approaches lead to test sets that distinguish a significant number of all mutants and provide high code coverage. Four Fortran programs and Mothra [17] tool were used.

Slightly different approach to mutants' sampling was proposed in [18]. Scholive, Beroulle and Robach proposed to choose a subset of mutants generated for each mutation operator. For four programs used in the experiment a subset containing 10% of mutants was created. The testing results for the complete set of mutants and the subset were better (more mutants were killed) than testing with randomly chosen mutants.

Offutt, Rothermel and Zapf in [19] were examining constrained mutation (some mutation operators were ignored). Ten Fortran programs (with 10 to 48 lines of code) were used in the experiment. Analyzing the results of this experiments authors proposed the category of mutation operators (named E). However this category contained only 5 operators, for ten tested programs the minimal factor of killed mutants was 98.67%, the average was 99.51%. Selective mutation decreased the number of mutants by 44% in average.

Using the results of the above described experiments and performing others experiments Mresa and Bottaci proposed in [21] the set of efficient mutation operators

Another approach to the mutant reduction problem was proposed by Patrick, Oriol and Clark in [22]. They propose to use static analysis to reduce mutants.

In [23] we examined randomly sampling mutants in object (Java) programs. Our experiment shows that randomly sampling 60% or 50% of mutants in Java programs can significantly reduce the cost of testing with acceptable mutation score and code coverage. Also these subsets of mutants were effective in detecting errors introduced by users. In [38] we described experiments with the reduction of mutants generated for by a mutation operator. This reduction is based on rules proposed by Scholive, Beroulle and Robach in [18].

Interesting approach to reduce the cost of mutation testing is proposed by Polo, Piattini and Garcia-Rodrıguez [24]. Reduction of mutation costs applied to C# programs are presented by Derezinska in [25, 26].

The above listed approaches to reduce the costs of mutation were aimed at reduction the number of mutants. Effective technique to improve the efficiency of mutation testing, without losing effectiveness, is parallel execution, where mutants and tests are executed in parallel processors, reducing the total time needed to perform mutation analysis. Mateo and Usaola in [27] present a study of this technique adapted to current technologies.

4 Experiment

The goal of the experiment was to explore the selective reduction of mutants **generated by the mutation operator.** Our experiments were conducted in the Eclipse environment. MuClipse [11] and CodePro [28] plugins were used for the mutation testing. Two special tools: Mutants Remover [29] and Console Output Analyzer [29] were designed and implemented especially for this experiment. Eight Java classes (listed in Table 1) were tested. For these classes 53 to 556 mutants were generated.

Table 1. Tested classes

class	Project	source	Number of methods	Lines of code	Number of mutants/equivalent mutants
Recipe	CoffeeMaker	[30]	14	84	138/15
CoffeeMaker	CoffeeMaker	[30]	8	102	285/17
Money	CodePro JUnit Demo	[28]	14	59	53/4
MoneyBag	CodePro JUnit Demo	[28]	17	114	54/6
Element	MapMaker	[31]	10	80	380/20
Board	NetworkShipBattle	[32]	12	123	270/3
Wall	jet-tetris	[33]	7	79	290/19
Stack	javasol	[34]	26	176	556/30

4.1 Experiment Method

For each class, being the subject of our experiment, firstly all mutants were generated. Secondly, the test cases killing these mutants were generated using JUnit, part of CodePro plugin. Console Output Analyzer [29] was identifying test cases not killing mutants. The identification and elimination of equivalent mutants, based on the analysis of source code of the original program and its mutants was time consuming and was made without any tool support. The tester had to construct several test cases especially for some nonequivalent mutants to obtain an adequate test set. The number of test cases generated automatically by CodePro was only 28.78% so quite a lot of time was spend on constructing test cases “manually”.

The initial set of all generated mutants was reduced by several methods. In this chapter the experimental results **of omitting mutants for arbitrary chosen mutation operator are presented** while in [23] we showed the results of randomly reducing the sets of mutants .

In the next step test cases “killing” all mutants in the set were produced. Firstly the CodePro generator was generating test cases and Console Output Analyzer [29] was identifying test cases not killing mutants. For the live mutants the test cases prepared for the whole set of mutants were used.

The sets of mutants and theirs test cases were next **evaluated using “killed” mutants factor**. We assumed arbitrary that test cases killing **95%** of all mutants are **adequate**.

In **selective mutation** the number of mutants was decreased by eliminating some of operators used in mutants' generation. For each of the tested class (listed in Table 1) operators generating the greatest number of mutants were identified (shown in Table 2). These operators are following: AOIS (Arithmetic Operator Insertion, Short-cut), ROR (Relational Operator Replacement), LOI (Logical Operator Insertion), AORB (Arithmetic Operator Replacement, Binary), COI (Conditional Operator Insertion), AOIU (Arithmetic Operator Insertion, Unary). In other studies [39], on other Java programs, using other mutation tool the operator AOIS also produced the most mutants (39.5% of the total) followed by ROR (14.15%), LOI (12.4%) and COI (11.6%).

In next step sets of mutants after the elimination of one of these operators were created. If the testing using such a set of mutants was satisfying the above given criterion we were trying to eliminate next operator, thus increasing the reduction of this set' cardinality. In Table 3 the average reduction of mutants after the elimination of selected operators is presented.

Table 2. Mutation operators generating the greatest number of mutants for tested classes

Klasa/Operator	AOIS	ROR	LOI	COI	AORB	AOIU
Recipe	50	27	15	7	0	10
CoffeeMaker	84	7	37	21	36	9
Money	14	10	4	3	12	3
MoneyBag	8	27	3	7	0	2
Element	170	39	54	22	60	14
Board	106	28	35	30	16	14
Wall	158	20	48	17	20	13
Stack	220	16	89	7	60	81
Total	*810*	*174*	*285*	*114*	*204*	*146*

Table 3. Reduced mutants after eliminating mutants from Table 3

Class/Operator	AOIS	ROR	LOI	COI	AORB	AOIU
Recipe	36.23%	19.57%	10.87%	5.07%	0%	7.25%
CoffeeMaker	29.47%	2.46%	12.98%	7.37%	12.63%	3.16%
Money	26.42%	18.87%	7.55%	5.66%	22.64%	5.66%
MoneyBag	14.81%	50%	5.56%	12.96%	0%	3.70%
Element	44.74%	10.26%	14.21%	5.79%	15.79%	3.68%
Board	39.26%	10.37%	12.96%	11.11%	5.93%	5.19%
Wall	54.48%	6.90%	16.55%	5.86%	6.90%	4.48%
Stack	39.57%	2.88%	16.01%	1.26%	10.79%	14.57%
average	*35.62%*	*15.16%*	*12.09%*	*6.89%*	*10.67%*	*5.96%*

4.2 Results of Experiment

Killed mutants factors for **selective** elimination of mutants for some operators (AOIS, ROR, LOI, COI, AORB, AOIU) are presented in Table 4. The values are impressive. In 43 out of 48 cases the killed factor was greater than 95% and in 33 cases was even 100% compared to the factor for the whole set of mutants. Only the elimination of mutants for operator AOIS appeared to be unsatisfactory .

Table 4. Killed mutants factors in selective mutations

class/omitted operator	AOIS	ROR	LOI	COI	AORB	AOIU
Recipe	91.87%	100%	100%	100%	100%	100%
CoffeeMaker	99.25%	100%	100%	100%	100%	100%
Money	95.92%	97.96%	100%	100%	91.84%	100%
MoneyBag	100%	85.42%	100%	100%	100.%	100%
Element	81.67%	98.61%	99.72%	100%	100%	100%
Board	99.25%	100%	100%	100%	100%	100%
Wall	98.15%	100%	100%	100%	100%	100%
Stack	92.02%	99.81%	100%	100%	98.48%	99.81%
average	*94.77%*	*97.72%*	*99.97%*	*100%*	*98.79%*	*99.98%*

Such high values of killed factor in selective mutations can be easily explained. It often happens that a test case is able to "kill" several mutants and omitting one of them will not decrease the factor for the complete set of mutants.

In Code 1 a test case named *equals_7* is shown. This test case, dedicated to the method *equals* of class *Money* (Code 2), checks if the method gives correct results if compared values are of the same type but one is higher. This test case is able to kill the mutant ROR_2 shown in Code 3 (relational operator „==" is changed into „>=" in line 46). The same test case is also able to kill mutant COR_1 (Code 4), conditional operator „&&" is changed into „||" in line 46. If only one operator ROR or COR will be omitted the test case *equals_7* still will be useful (killing other mutant) and the killed mutant factor will not decrease (compared to the whole set of mutants).

```
public void testEquals_7() {
  Money money = new Money(15, "USD");
  Object anObject = new Money(5, "USD");

  boolean result = money.equals(anObject);
  assertEquals(false, result);
}
```

Code 1 Test case equals_7 killing mutants ROR_2 and COR_1

```
39) public boolean equals(Object anObject) {
40)   if (isZero())
41)      if (anObject instanceof IMoney)
42)         return ((IMoney)anObject).isZero();
43)    if (anObject instanceof Money) {
44)       Money aMoney = (Money)anObject;
45)   return aMoney.currency().equals(currency())
46)      && amount() == aMoney.amount();
47)   }
48)   return false;
49) }
```

Code 2 Method *equals* of class *Money*

```
39) public boolean equals(Object anObject) {
40)   if (isZero())
41)      if (anObject instanceof IMoney)
42)         return ((IMoney)anObject).isZero();
```

```
43)     if (anObject instanceof Money) {
44)        Money aMoney = (Money)anObject;
45)     return aMoney.currency().equals(currency())
46)        && amount() >= aMoney.amount();
47)     }
48)     return false;
49)  }
```

Code 3 Mutant ROR_2 of method *equals* class *Money*

```
39)  public boolean equals(Object anObject) {
40)    if (isZero())
41)       if (anObject instanceof IMoney)
42)          return ((IMoney)anObject).isZero();
43)     if (anObject instanceof Money) {
44)        Money aMoney = (Money)anObject;
45)     return aMoney.currency().equals(currency())
46)        || amount() == aMoney.amount();
47)     }
48)     return false;
49)  }
```

Code 4 Mutant COR_1 of method *equals* in classy *Money*

The results obtained for the majority of selectively reduced sets of mutants were satisfying assumed by us criterion that the value of killed mutant factor is at least 95%. Based on the reasoning presented above we were reducing the set of mutants omitting **more than one operator**. We started with omitting the operator with the high average level of mutants reduction (Table 3). If the average value for killed mutant factor was greater than 95% we were omitting successive operators. The changes of the killed factor for some classes are presented in Fig. 1 and in Table 5 all results of this experiments are presented. Operators ROR, LOI, AORB, COI and AOIU were successively omitted . Even after the elimination of 4 operators the killed factor was close to 95%. Operator AOIS provides the highest level of reduction (Table 3) but if it was omitted the killed factor was immediately below assumed by us level of 95%.

a)

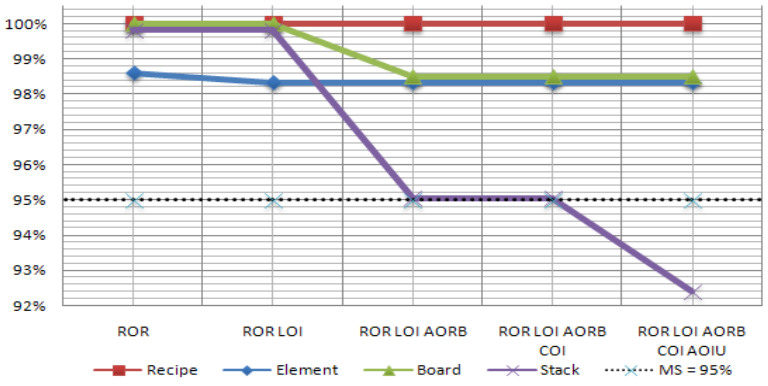


Fig. 1. Killed factor after omitting successive operators (classes Recipe, Money, Board, Stack)

Table 5. Killed mutants factors for SM subsets omitting more than one operator

class/omitted operators	ROR, LOI	ROR, LOI, AORB	ROR, LOI, AORB, COI	ROR, LOI, AORB, COI, AOIU
Recipe	100.00%	100.00%	100.00%	100.00%
CoffeeMaker	100.00%	100.00%	100.00%	100.00%
Money	97.96%	85.71%	85.71%	85.71%
MoneyBag	85.42%	85.42%	85.42%	85.42%
Element	98.33%	98.33%	98.33%	98.33%
Board	100.00%	98.50%	98.50%	98.50%
Wall	100.00%	100.00%	98.89%	98.89%
Stack	99.81%	95.06%	95.06%	92.40%
average	*97.69%*	*95.38%*	*95.24%*	*94.91%*

5 Conclusions

Experimental research has shown the mutation testing to be very effective in detecting faults [6,23,35,36,37], unfortunately it is computationally expensive so some researchers propose parallel execution of tests [27], others are constraining the sets of mutants. The goal of our research was to examine the reduction some mutation operators including class operators in Java programs. The previous experiments were made with structural languages, mainly Fortran (e.g. [19]), some also with object languages (e.g. C# [25,26]). Our experiment, described in section 4, shows that omitting mutants generated for chosen mutation operator (regular and class level) in Java programs can significantly reduce the cost of testing with acceptable mutation score. The experiments reported in this chapter needed a lot of effort so only 8 Java classes were tested. The number of programs used in other experiments on mutation' subset were similar. It is difficult to know if 8 classes is sufficiently large sample from which to generalize and so similar studies on larger sets of classes will be useful. All the results of this study have been obtained using the set of mutation operators present in MuClipse and cannot be applied directly to mutation systems that use different operators. Efficiency relationships will, nonetheless, be present between any set of operators.

References

1. Jia, Y., Harman, M.: An Analysis and Survey of the Development of Mutation Testing. Technical report TR-09-06, Crest Centre, Kong's College London, `http://www.dcs.kcl.ac.uk/pg/jiayue/repository/TR-09-06.pdf` (accessed 2013)
2. Mutation repository, `http://crestweb.cs.ucl.ac.uk/resources/mutation_testing_repository/` (accessed XII 2013 but modified VII 2011)
3. Bashir, B.M., Nadeem, A.: Object Oriented Mutation Testing: A Survey. IEEE: 978-1-4673-4451-7/12 (2012)
4. Mathur, A.P.: Mutation Testing. In: Marciniak, J.J. (ed.) Encyclopedia of Software Engineering, pp. 707–713 (1994)

5. Schuler, D., Zeller, A.: (Un-)covering equivalent mutants. In: Proc. ICST, pp. 45–54 (2010)
6. Voas, J.M., Miller, K.W.: Software testability: the new verification. IEEE Softw. 12(3), 17–28 (1995)
7. Grun, B., Schuler, D., Zeller, A.: The impact of equivalent mutants. In: Proceedings of the 4th International Workshop on Mutation Testing (2009)
8. Hierons, R., Harman, M., Danicic, S.: Using program slicing to assist in the detection of equivalent mutants. Softw. Test. Verif. Rel. 9(4), 233–262 (1999)
9. Offutt, A.J., Craft, W.M.: Using compiler optimization techniques to detect equivalent mutants. Softw. Test. Verif. Rel. 4(3), 131–154 (1994)
10. Offutt, A.J., Pan, J.: Automatically detecting equivalent mutants and infeasible paths. Softw. Test. Verif. Rel. 7(3), 165–192 (1997)
11. MuClipse, `http://muclipse.sourceforge.net/index.php` (accessed 2012)
12. Madeyski, L., Radyk, R.: Judy - A Mutation Testing Tool for Java. IET Software 4(1), 32–42, doi:10.1049/iet-sen.2008.0038
13. Jumble, `http://jumble.sourceforge.net/index.ht` (accessed 2010)
14. Acree, A.T.: On mutation. Ph.D. thesis, Georgia Institute of Technology, Atlanta, Georgia (1980)
15. Budd, T.A.: Mutation analysis of program test data. Ph.D. thesis, Yale Univesity, New Haven, Connecticut (1980)
16. Mathur, A.P., Wong, W.E.: Reducing the cost of mutation testing: an empirical study. J. Syst. Softw. 31(3), 185–196 (1995)
17. DeMillo, R.A., Guindi, D.S., King, K.N., McCracken, W.M., Offutt, A.J.: An extended overview of the Mothra software testing environment. In: Proc. of the Second Workshop on Software Testing, Verification, and Analysis, pp. 142–151 (July 1988)
18. Scholive, M., Beroulle, V., Robach, C.: Mutation Sampling Technique for the Generation of Structural Test Data. In: Proc. of the Conf. on Design, Automation and Test in Europe, March 7-11, vol. 2, pp. 1022–1023 (2005)
19. Offutt, J., Rothermel, G., Zapf, C.: An experimental determination of sufficient mutation operators. ACM Trans. on Soft. Eng. and Methodology 5(2), 99–118 (1996)
20. DeMillo, R.A., Offutt, A.J.: Constraint-based automatic test data generation. IEEE Trans. on Soft. Eng. 17(9), 900–910 (1991)
21. Mresa, E.S., Bottaci, L.: Efficiency of mutation operators and selective mutation strategies: An empirical study. Soft. Testing, Ver. and Rel. 9(4), 205–232 (1999) ISSN: 09600833
22. Patrick, M., Oriol, M., Clark, J.A.: MESSI: Mutant Evaluation by Static Semantic Interpretation. In: 2012 IEEE Fifth Int. Conf. on Software Testing, Verification and Validation, pp. 711–719 (2012), doi:10.1109/ICST.2012.161
23. Bluemke, I., Kulesza, K.: Reduction of computational cost in mutation testing by sampling mutants. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 41–51. Springer, Heidelberg (2013), doi:10.1007/978-3-319-00945-2_4
24. Polo, M., Piattini, M., Garcia-Rodrıguez, I.: Decreasing the cost of mutation testing with second order mutants. Softw. Test. Verif. Reliab. 19, 111–131 (2009)
25. Derezińska, A., Rudnik, M.: Quality Evaluation of Object-Oriented and Standard Mutation Operators Applied to C# Programs. In: Furia, C.A., Nanz, S. (eds.) TOOLS 2012. LNCS, vol. 7304, pp. 42–57. Springer, Heidelberg (2012)
26. Derezińska, A.: A Quality Estimation of Mutation Clustering in C# Programs. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J., et al. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 119–129. Springer, Heidelberg (2013)

27. Mateo, P.R., Usaola, M.P.: Parallel mutation testing. Softw. Test. Verif. Reliab. 23, 315–350 (2013)
28. CodePro JUnit Demo, https://developers.google.com/java-dev-tools/codepro/doc/features/junit/CodeProJUnitDemo.zip (accessed 2012)
29. Kulesza, K.: Mutation testing computational cost reduction using mutants sampling and selective mutation (in polish). M.Sc. thesis, Institute of Computer Science, Warsaw University of Technology (September 2012)
30. http://agile.csc.ncsu.edu/SEMaterials/tutorials/coffee_maker (accessed April 2012)
31. Godlewski, Ł.: MapMaker. Institute of Computer Science, Warsaw University of Technology (2008) (unpublished)
32. Suchowski, J.: Network game – NetworkShipsBattle. Institute of Computer Science, Warsaw University of Technology (2010) (unpublished)
33. jet-tetris, http://sourceforge.net/projects/jet-tetris (accessed 2012)
34. javasol, http://sourceforge.net/projects/javasol (accessed May 2012)
35. Bluemke, I., Kulesza, K.: A Comparison of Dataflow and Mutation Testing of Java Methods. In: Zamojski, W., Kacprzyk, J., Mazurkiewicz, J., Sugier, J., Walkowiak, T. (eds.) Dependable Computer Systems. AISC, vol. 97, pp. 17–30. Springer, Heidelberg (2011)
36. Frankl, P.G., Weiss, S.N., Hu: All-uses versus mutation testing: an experimental comparison of effectiveness. J. Syst. Softw. 38(3), 235–253 (1997)
37. Andrews, J.H., Briand, L.C., Labiche, Y.: Is mutation an appropriate tool for testing experiments? In: Proc. ICSE, pp. 402–411 (2005)
38. Segura, S., Hierons, R.M., Benavides, D., Ruiz-Cortés, A.: Mutation testing on an object-oriented framework: An experience report. Information and Software Technology 53, 1124–1136 (2011)

An Approach for Planning and Analysis of the Sewage Sanitary Networks Using Some Calculation Formulas and Computer Simulation*

Lucyna Bogdan, Grażyna Petriczek, and Jan Studziński

Systems Research Institute, Polish Academy of Science
01-447 Warsaw, ul. Newelska

Abstract. The hydraulic calculations are carried out using nomograms, which are the charts connecting diameters, flow rates, hydraulic slopes and average flow velocities. In traditional planning of sewage networks the appropriate hydraulic values are read from the nomogram chart tables. In the paper another way of executing of hydraulic calculations is presented. The numerical solutions of nonlinear equations describing the phenomena of sewage flows are used. The presented method enables the quick analysis of sewage net parameters and opens the possibility of sewage network computer simulation.

Keywords: mathematical modeling of sewage network, hydraulic parameter of canal.

1 Introduction

Planning of communal sewage networks is a very complex task because of the complexity of mathematical equations describing the wastewater flows in network canals and because of the great variety of networks types. In case of water networks which are pressure systems their main parameters are water flows and water pressures whose values are dependent on pipe diameters and from water pressures produced in the pump stations located on the networks. Against it the wastewater networks are gravitational systems whose main hydraulic parameters are sewage flows and filling heights of the canals and the factors that decide about their values are the diameters, slopes and profiles of canals. The classic approach of planning of sewage systems consists in using of the co called nomograms which are some diagrams combining together the canal diameters, slopes and filling heights as well as the sewage flow intensities and velocities. The wanted values of these parameters are picked off from the nomograms which are results of former calculations made with the formulas commonly used for computing the sewage networks (formulas of Chezy, Colebrooke-White and of Manning) [1], [3], [4], [12]. More advanced approach for planning the communal sewage systems means the use of hydraulic models of wastewater net-

* The paper is a result of the research project No N N519 6521 40 financed by the Polish National Center of Scence NCN.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_10,

works like the MOSKAN software developed at the Systems Research Institute what requires some informatics knowledge. The first and classical approach of planning of sewage systems is very mechanical and the second one is more complicated.

In the paper an indirect approach to calculate the hydraulic parameters of wastewater networks is proposed in which an algorithm for a rather simple numerical solution of nonlinear equations resulted from the main hydraulic formulas and rules describing the networks is presented. The method applied in the algorithm makes possible fast analysis of the main network parameters, i.e. the canal filling heights and the sewage flow velocities, and it enables this way the fast and simple simulation of the investigated sewage system. The algorithm presented has been used for simulation of an exemplary wastewater network of sanitary type. Changing the values of intensities of sewage inflows to the network in some chosen network nodes one can simply calculate new values of canal filling heights and sewage flow velocities in the canals connected with these nodes. The approach proposed enables also the understanding of the relations between different hydraulic parameters of sewage canals.

2 Basic Problems

For the considered algorithms of wastewater networks calculation the following basic assumptions are made:

- Only housekeeping or combined sewage nets are considered, divided into branches and segments by nodes.
- The nodes are the points of connection of several network segments or branches or the points of changing of network parameters as well as of location of sewage inflows to the network (sink basins, rain inlets, connecting basins). In the connecting nodes the flow balance equations and the condition of levels consistence are satisfied.
- It is assumed that the segments parameters such as shape, canal dimension, bottom slope or roughness are constant. Because of these assumptions all relations concern the steady state problem.
- The nets considered are of gravitational type.

Designing and analysis of sewage networks are connected with the following tasks:

1. Making hydraulic analysis of the network for known section crosses and for known canal slopes. In this case the calculation of filling heights of the canals as well as the calculation of flow velocities depending on the sewage flow rates must be done. These calculations are done for the respective net segments using the earlier received flow values.
2. Designing of new segments of the network. It concerns the case when the new segments of the network must be added to the existing ones. In this situation diameters and canals slopes must be chosen for the new canals. It is assumed that the sewage inflows are known.

3 Algorithms for the Calculation of Wastewater Networks

3.1 The Algorithm for Calculation of Canal Filling Heights and Flow Velocities

Presented below algorithm is used for analysis of work of sewage net. The algorithm presented requires the following data for its calculation:

- type of the network– housekeeping sewage net or combined sewage net
- structure of the network – numbers of segments and nodes and type of nodes
- maximal sewage inflow into particular sewage net nodes
- slows of canal bottoms and the canal dimensions.

The task of the algorithm is to determine the following values for given values of rate inflows Q_i:

- filling heights in each wastewater network segment
- flow velocity for each network segment.

The calculation scheme presented below is for the canals with circular section. The most important part of algorithm is the calculation of the filling heights H_i (or canal filling degree $x= H_i/d_i$) and the flow velocities v_i for each wastewater network segment (for given values of rate inflows Q_i in particular sewage net nodes).

Now the problem is to solve the nonlinear algebraic equations which are derived from the basic relations and hydraulic formulas.

1. From the Manning formula and taking into account the canal geometry one can obtain the following relations with x=H/d [1], [2], [3]:

$$\text{For } x \le 0{,}5 \qquad \beta \cdot F_1(x) - Q = 0 \tag{1a}$$

$$F_1(x) = \frac{(\varphi_1(x) - \sin(\varphi_1(x)))^{\frac{5}{3}}}{\varphi_1(x)^{\frac{2}{3}}} \tag{1b}$$

$$\varphi_1(x) = 2 \cdot \arccos(1 - 2 \cdot x) \tag{1c}$$

$$\text{For } x > 0{,}5 \quad \beta \cdot F_2(x) - Q = 0 \tag{2a}$$

$$F_2(x) = 2 \cdot \frac{(\pi - 0{,}5 \cdot \varphi_2(x) + 0{,}5 \cdot \sin(\varphi_2(x)))^{\frac{5}{3}}}{(\pi - 0{,}5 \cdot \varphi_2(x))^{\frac{2}{3}}} \tag{2b}$$

$$\varphi_2(x) = 2 \cdot \arccos(2 \cdot x - 1) \tag{2c}$$

$$\beta = 0{,}5 \cdot \frac{1}{n} \cdot (d)^{\frac{8}{3}} \cdot \left(\frac{1}{4}\right)^{\frac{5}{3}} \cdot J^{\frac{1}{2}} \tag{3}$$

where: H – filling height, φ – central angle, d – inside canal diameter, J – canal slope, n – roughness coefficient, Q – rate inflow, H/d - canal filling degree.

The β parameter in (3) depends on canal diameter d and on canal slope J and for the fixed diameter values and canal slopes it is constant. Solving equations (1a)–(2b) we obtain canal filling degree H/d as a function of flow rate Q. Equations (1a)–(2b) for calculating the canal filling degree are nonlinear and the standard numerical methods for solving nonlinear algebraic equations can be applied.

In order to determine the equation roots some conditions for parameter β and sewage flow Q must be fulfilled that will be discussed below.

For fixed network parameters like canal diameter d and canal slope J, equation $\beta \cdot F(x) - Q = 0$ has solutions depending on sewage flow Q.

Equation $\beta \cdot F(x) - Q = 0$ has the following roots:

a. For $x \in (0; 0.5>$ there is only one root and the following inequality must be fulfilled: $0 < Q \leq \pi \cdot \beta$. This inequality defines a values range for sewage flows Q for fixed canal diameters d and canal slopes J.

b. For $x \in (0.5; 1>$ equation $\beta \cdot F(x) - Q = 0$ has the following roots:

- one root for $x \in (0.5; 1)$ and $\pi \cdot \beta < Q < 2\pi \cdot \beta$ (Fig.1)
- two roots for $x \in (0.5; 1>$ and $2\pi \cdot \beta \leq Q < \beta \bullet \cdot 6.7586936$, whereas for $Q = 2\pi \cdot \beta$ there are $x_1 = 1$ and $x_2 = 0.81963$.

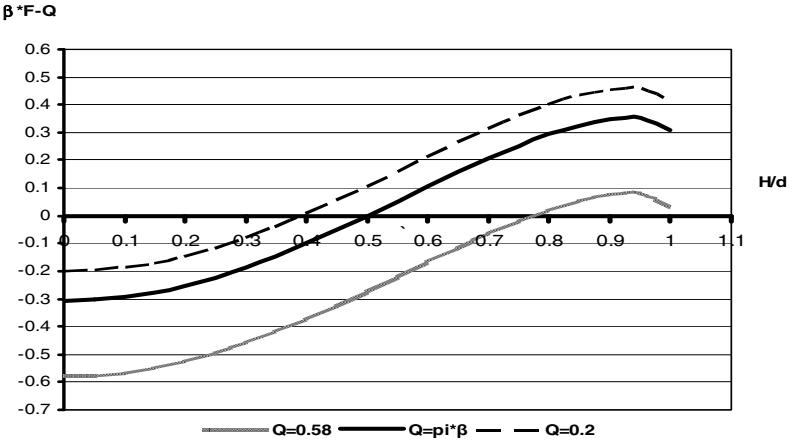


Fig. 1. Diagrams of function $\beta \cdot F(x) - Q$ for different values of Q in values range $(0; 2\pi \cdot \beta)$

This above analysis has been done for d=0.6, J=1% , n=0.013 and is presented in Fig.1.

For the fixed network parameters such as a canal diameter d and canal slope J the above relations let to decide what are the solutions for the given flow Q and whether the value of Q is not greater than the upper limit $\beta \bullet \cdot 6.7586936$, what means the lack of solutions. In such the case a change of one or of both of the fixed network parameters d and J must be considered. The result of the above relations is that the flow value Q depends on the parameter β.

The parameter β depends on the canal diameter d and on the canal slope J. The equation describing the dependence of canal filling on the flow in the range $(0; 2\pi \cdot \beta)$ has one solution in this range and that is why this range is relevant.

In Fig.2 the relation between the solution of equation $\beta \cdot F(x) - Q = 0$ and flow Q for d=0.6, J=1%, n=0.013 and $0 < Q < 2\pi \cdot \beta$ is graphically shown.

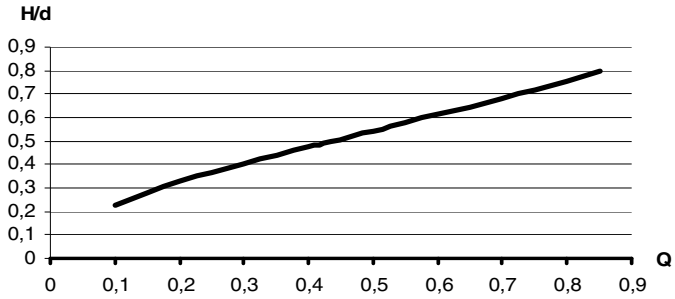


Fig. 2. Relation between the solution of $\beta \cdot F(x) - Q = 0$ and flow Q

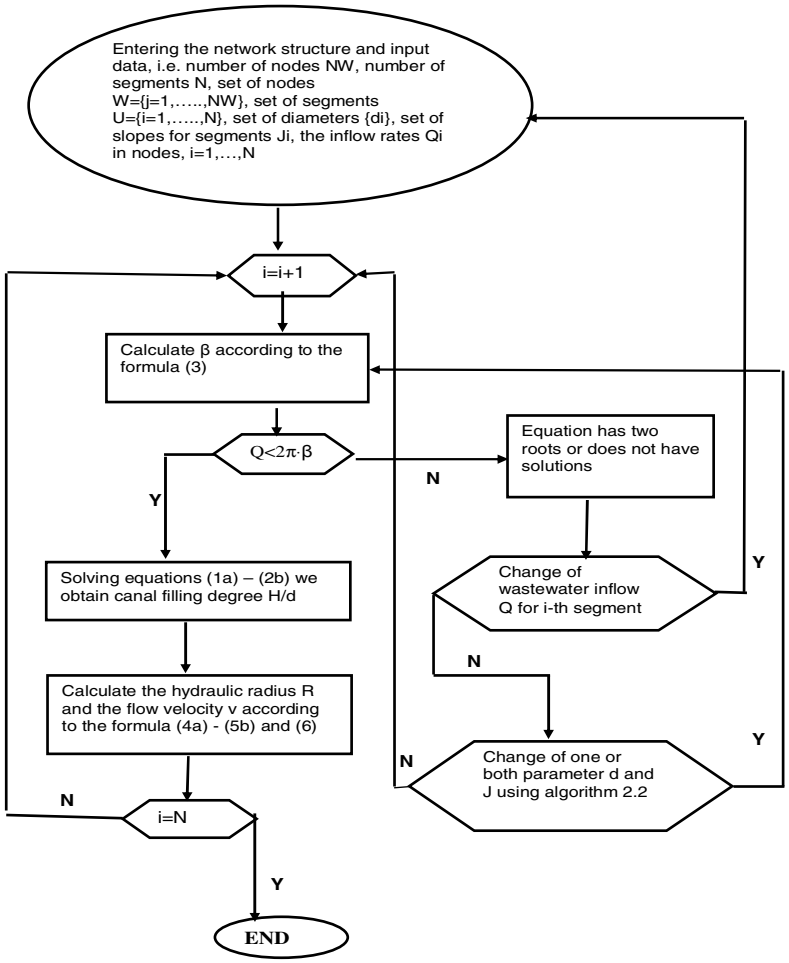


Fig. 3. Scheme of algorithm for calculation of canal filling degree and flow velocity

2. For canal filling degree H/d calculated above the hydraulic radius R should be determined according to the formula:

For H/d <= 0.5:
$$R = \frac{1}{4} d \left(1 - \frac{\sin \varphi}{\varphi} \right) \tag{4a}$$
$$\varphi = 2 \cdot \arccos \left(1 - 2 \cdot \frac{H}{d} \right) \tag{4b}$$

For H/d > 0.5:
$$R = \frac{d}{4} \left(\frac{\pi - 0.5\varphi + 0.5 \sin(\varphi)}{\pi - 0.5\varphi} \right) \tag{5a}$$
$$\varphi = 2 \cdot \arccos \left(2 \cdot \frac{H}{d} - 1 \right) \tag{5b}$$

3. The flow velocity should be calculated from:
$$v = \frac{1}{n} R^{\frac{2}{3}} \cdot J^{\frac{1}{2}} \tag{6}$$

Knowing the network geometry, i.e. slopes, shapes and diameters of canals as well as the wastewater inflows Q_i, one can calculate filling heights and flow velocities for each network canal. The calculations are carried out for each network segment beginning from the farthest one and going step by step to the nearest segment regarding the wastewater treatment plant. The algorithm scheme is shown in Fig. 3.

The whole network will be calculated once again with the wastewater inflows changed. Under assumption of constant sewage flows in the network segments the sewage system simulation can be executed for a sequence of time steps, for a couple of hours or days; by such the calculation the change of the wastewater inflows occurring with the time must be considered.

3.2 The Algorithm for Calculation of Canal Diameters d and Canal Slopes J for Given Flow Values Q

The conclusion is that the value of flow Q depends on the parameter β, which depends on the canal diameter d and on the bottom slope J. The equation describing the dependence of the filling degree from the flow has one solution in the interval (0; $2\pi \cdot \beta$) and this is the cause that this interval is relevant.

The calculation procedure shown below concerns the following cases:

- Flow Q exceeds the upper boundary of values domain for β•·6.7586936; then a change of values for given canal diameters d and slopes J have to be considered.
- New segments must be added to the existing network; then the diameters and slopes must be defined for the new canals under the assumption that the sewage inflows Q into the canals have been forecasted and they are known.

In both cases while calculating diameters and slopes for the new canals for given flows Q the inequality $2\pi \cdot \beta - Q > 0$ has to be considered. The fulfilling of the inequality qwarrants the existence of only 1 solution of the equation describing the dependence the filling degree x from the canal flow Q. The calculation procedure consists of the following steps which are realized for the forecasted and fixed flow values Q:

Step 1. Determination of canal slope value J. The value can be determined according to the existing technical standards or calculated regarding the relations for minimal slopes which are known from literature [4], [5], [6], [12].

Step 2. Solution of the following equation:

$$\varsigma \cdot d^{\frac{8}{3}} - Q = 0 \qquad \varsigma = \frac{\pi}{n} \cdot \left(\frac{1}{4}\right)^{\frac{5}{3}} \cdot J^{\frac{1}{2}} \tag{7}$$

a) For $J = \frac{a}{d}$

$$\alpha_1 \cdot d^{\frac{13}{6}} - Q = 0 \qquad \alpha_1 = \frac{\pi}{n} \cdot \left(\frac{1}{4}\right)^{\frac{5}{3}} \cdot a^{\frac{1}{2}} \tag{8a}$$

b) For J ensuring canal self purification:

$$\alpha_2 \cdot d^{\frac{13}{6}} - Q = 0 \qquad \alpha_2 = \frac{2\pi}{n} \cdot \left(\frac{1}{4}\right)^{\frac{5}{3}} \cdot \left(\frac{\tau_{min}}{1{,}1106 \cdot \rho}\right)^{\frac{1}{2}} \tag{8b}$$

c) For the limiting slope J

$$\alpha_3 \cdot d^{\frac{5}{2}} - Q = 0 \qquad \alpha_3 = \frac{\pi}{n} \cdot \left(\frac{1}{4}\right)^{\frac{5}{3}} \cdot \left(3{,}778 \cdot 10^{-3}\right)^{\frac{1}{2}} \tag{8c}$$

If a solution d_* of the equation exists, then inequality $\varsigma \cdot d^{\frac{8}{3}} - Q > 0$ is valid for all values $d > d_*$. If canal slope J has been calculated from relations (a)–(c) and now a value d greater than d_* will be taken into account, then one shall pass to Step 1 and the canal slope must be calculated again. If a solution of equation (7) does not exists then one shall return to Step 1, change the value J and solve once again equation (7).

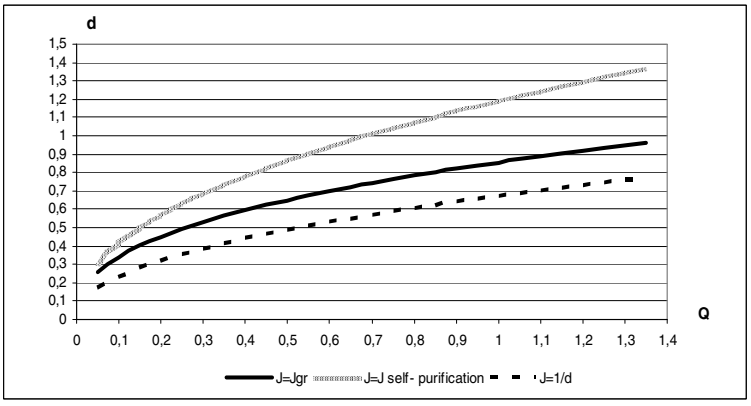


Fig. 4. Relations between canal diameter d and canal flow Q for characteristic canal slopes J

4 Simulation of Wastewater Network

The considered algorithm has been tested on an exemplary housekeeping network consisting of 27 nodes connected by 26 segments (Fig. 5).

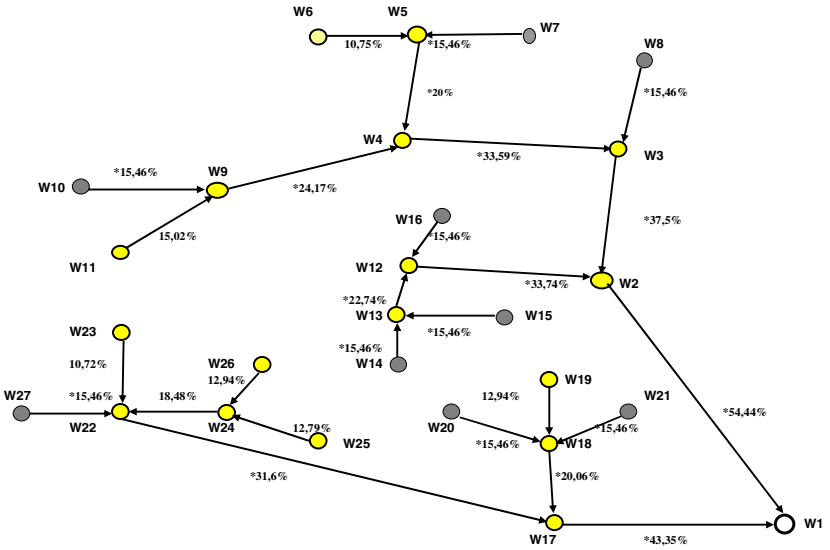


Fig. 5. Structure of the sewage net and simulation results

Table 1. The results of hydraulic computations for the exemplary net shown in Fig. 5

Upper node	Lower node	Segment	input flows in node	flows in segments Q [dm3/s]	H/d [%]	v [m/s]	H/d [%] MOSKAN	v [m/s] MOSKAN
W6	W5	1	0,56	0,56	10,72	0,309	11	0,29
W7	W5	2	0,31	0,31	8,09	0,259	8	0,26
W5	W4	3	0,27	1,14	15,08	0,383	15	0,38
W10	W9	4	0,36	0,36	8,69	0,271	9	0,27
W11	W9	5	1,13	1,13	15,02	0,382	14,6	0,39
W9	W4	6	0,64	2,13	20,48	0,460	20	0,46
W4	W3	7	0,64	3,91	27,78	0,549	28	0,55
W8	W3	8	0,11	0,11	4,98	0,189	5	0,19
W3	W2	9	0,1	4,12	28,53	0,557	29	0,56
W14	W13	10	0,11	0,11	4,98	0,189	5	0,19
W15	W13	11	0,32	0,32	8,22	0,261	8	0,26
W13	W12	12	0,23	0,66	11,59	0,325	12	0,33
W16	W12	13	0,24	0,24	7,17	0,240	7	0,24
W12	W2	14	1,86	2,76	23,29	0,497	23	0,49
W2	W1	15	0,73	7,61	39,42	0,661	39	0,66
W23	W22	16	0,56	0,56	10,72	0,309	11	0,29
W27	W22	17	0,4	0,4	9,13	0,280	9	0,27
W25	W24	18	0,81	0,81	12,79	0,346	13	0,36

Table 1. *(continued)*

W26	W24	19	0,83	0,83	12,94	0,348	13	0,38
W24	W22	20	0,09	1,73	18,48	0,433	18	0,43
W22	W17	21	1,53	4,22	28,89	0,561	29	0,56
W19	W18	22	0,83	0,83	12,94	0,348	12	0,38
W20	W18	23	0,3	0,3	7,97	0,256	7	0,26
W21	W18	24	0,19	0,19	6,42	0,223	6	0,22
W18	W17	25	0,22	1,54	17,46	0,419	18	0,49
W17	W1	26	0,57	6,33	35,70	0,629	36	0,63
W1	Sewage plant			13,94				

The net has got 15 input nodes (W_6, W_7, W_8, W_{10}, W_{11}, W_{14}, W_{15}, W_{16}, W_{19}, W_{20}, W_{21}, W_{23} , W_{25}, W_{26}, W_{27}) and 1 output node W_1. Other nodes constitute the connections between different segments of the network [10]. The arrows in Fig. 5 show the sewage flow direction. The sewage flow rates values for the input nodes are given. The flow rates in the connection nodes should be calculated according to the balance equation. For the respective segments the diameters d=0,2 and the canal slopes J=0,5‰ are given. For such a structure of the net the fillings H/d and the velocities of flows v in respective segments are calculated. The conclusion is that for these values of sewage rate flows and for the given values of geometric parameters (diameters and canal slopes) the heights of filling are lower than the half of canal diameters. So there is a possibility of increasing of the input flows in some sewage nodes.

The calculations results are shown in Table 1. The simulation was done by changing the input flows in these nodes for which the flow values in the connected segments are less than 0,5. The results of the simulation are shown also in Fig. 5. On the arches of the net there are marked the calculated values of filling degrees H/d for the appropriate net segments. The values marked by the star concern the filling degrees calculated for the replaced flow values in the nodes W_7, W_8, W_{10}, W_{14}, W_{15}, W_{16}, W_{20}, W_{21}, W_{27}. The conclusion is that the change of input flow values in the chosen nodes causes the significant changes of the filling heights in the suitable net segments.

The results obtained are very similar to the ones received using the MOSKAN software. It means in our opinion that our approach is more reliable and dependable than this classical one using the nomograms and not less reliable than the modern and more complicated approach using hydraulic models of the sewage networks.

5 Conclusions

In the paper a new practical approach for computing sanitary sewage networks is proposed that differs from the approaches commonly used in the today's practice of sewage nets operation. The standard and mostly applied method of sewage networks calculation uses the nomograms which enable to calculate in pure mechanical way the

basic parameters of the designed nets such as diameters and canal slopes on the base of estimated sewage inflow values. The nomogram schemes have to be drawn before the process of network designing is started and their drawing occurs on the base of appropriate hydraulic equations and relations. The received results of sewage network calculation are approximate and depend on the quality of the schemes.

The modern approach in this field consists of applying advanced computer programs like SWMM [7, 8, 13], MIKE URBAN [14] or MOSKAN [9] which use in their computations hydraulic models of sewage networks. This approach requires an advanced computer knowledge from the program users and although the programs mentioned are already commonly in use on the universities then there is lack of their applications in the waterworks. The next obstacle in using this software in operational practice in the waterworks for designing the sewage networks is the necessity of having their right calibrated hydraulic models [10]. To calibrate the models a GIS system to generate the numerical map of the network and a properly dense monitoring system to collect the measurements data have to be installed on the sewage net what means expensive investments. The most of Polish waterworks are municipal enterprises and they have not enough money for buying these costly systems.

It seems that the approach for designing the sanitary sewage networks presented in the paper being an indirect method between the standard and modern ones can be currently an ideal tool for computing such networks for it owns the advantages of these both approaches and it has not their drawbacks. It uses the analytical relations concerning the hydraulics and geometry of sewage networks and it transform them to nonlinear equations from which depending on the requirements the of canal fillings and sewage speeds or canal diameters and slopes can be directly calculated. The analysis of the equations performed enables the determination of the available maximal sewage inflows going to the network nodes. In this way the calculations can be done quickly and exactly avoiding the using of the complicated hydraulic model of the sewage net.

In the presented calculation example the results received by means of the approach proposed have been compared with the results obtained with professional software MOSKAN [11] that is based on the hydraulic model SWMM [13] similar to MIKE URBAN software developed by DHI [14]. The parameter values of the sewage net calculated are the same for both cases. It means that the indirect approach presented is not less exact than the modern approach and at the same time it is much more simple and fast than this modern one and it is more exact and handy than the classical approach.

The computational example presented is rather simple but the approach proposed and the computer program developed for it can be used unproblematic also for modeling and designing more complex municipal sewage systems.

References

1. Biedugnis, S.: Metody informatyczne w wodociągach i kanalizacji. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa (1998)
2. Bogdan, L., Petriczek, G., Studziński, J.: Mathematical modeling and computer aided planning of communal sewage networks. JAMRIS, Industrial Research Institute for Automation and Measurements “PIAP”, Warsaw (2013) (in print)

3. Błaszczyk, W., Stamatello, H., Błaszczyk, P.: Kanalizacja. Sieci i pompownie. Tom I. Arkady, Warszawa (1983)
4. Chudzicki, J., Sosnowski, S.: Instalacje kanalizacyjne. Wydawnictwo Seidel-Przywecki, Warszawa (2004)
5. Kwietniewski, M., Nowakowska–Błaszczyk, A.: Obliczenia hydrauliczne kanałów ściekowych na podstawie krytycznych natężeń stycznych. Nowa Technika w Inżynierii Sanitarnej – Wodociągi i Kanalizacja, Warszawa (1981)
6. Puchalska, E., Sowiński, N.: Wymiarowanie kanałów ściekowych metodą krytycznych naprężeń stycznych. Ochrona Środowiska (3-4) (1984)
7. Rossman, L.: Storm Water Management Model (SWMM) – User's manual, Version 5.0.022 (2012), http://www.epa.gov/nrmrl/wswrd/wq/models/swmm/
8. Saegrov, S.: Care-S - Computer Aided Rehabilitation for Sewer and Storm Water Networks. IWA Publishing, Alliance House, London (2005)
9. Służalec, A., Studziński, J., Ziółkowski, A.: Optimization of sewerage structure in the integrated system for sewage design, management and revitalization - MOSKAN. In: Wittmann, J., Mueller, M. (hrsg.) Simulation in Umwelt- und Geowissenschaften. ASIM-Mitteilung AM, vol. 146, pp. 203–210 (2013)
10. Służalec, A., Studzinski, J., Wójtowicz, P., Ziółkowski, A.: Erstellung des hydraulischen Modells eines kommunalen Abwassernetzes und dessen Kalibrierung anhand echter Daten. In: Thinh, N.X. (hrsg.) Modelierung und Simulation von Ökosystemen. Umweltinformatik, Reihe, Shaker Verlag, Aachen (2013) (in print)
11. Służalec, A., Studziński, J., Ziółkowski, A.: Rechnerunterstütze Planung von kommunalen Abwassernetzen mittels des hydraulischen Models und statischer Optimierung. In: Thinh, N.X. (hrsg.) Workshop Koelpinsee 2012, pp. 123–133. Shaker Verlag, Aachen (2013)
12. Wartalski, A., Wartalski, J.: Projektowanie hydrauliczne rurociągów z tworzyw sztucznych. Ochrona Środowiska 76(1), 19–24 (2000)
13. http://www.epa.gov/nrmrl/wswrd/wq/models/swmm/
14. http://www.mikebydhi.com/Products/Cities/MIKEURBAN.aspx

Mathematical Model of Task Scheduling in Educational Cloud

Agata Brzozowska and Jerzy Greblicki

Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
11-17 Janiszewskiego Street, 50-372 Wrocław, Poland
{agata.brzozowska,jerzy.greblicki}@pwr.wroc.pl

Abstract. In this paper we present problems connected to cloud computing in particular scheduling of virtualization tasks. We consider different types of tasks – computational, virtual desktops and servers. Cloud computing at University is on the one hand interesting tool for researchers and on the other very interesting topic to research. We propose a mathematical model for virtual machines scheduling in cloud computing systems.

Keywords: Cloud computing, scheduling, mathematical model.

1 Introduction

In this paper we present problem of task scheduling which is considered one of the main parts of Cloud Computing (CC). We understand this process as mapping users tasks to appropriate resources to execute. The process of assignment is transparent for end point users. To achieve this goal it is necessary to use virtual machines (VM). Task scheduling is therefore critical for the whole system. Scheduling algorithms for CC systems are discussed in literature [2]. Because most authors use mathematical model without consideration of desktop virtualization issues we propose a new model for tasks scheduling and virtual machines deployment for business and educational purposes.

2 Cloud Computing

Cloud computing is a relatively new processing model. In the last few years, the term Cloud is increasingly popular in both the commercial use and as a scientific term [3].

Cloud computing allows to share resources, software, and information over the Internet. But it is sharing services that makes cloud completely new and innovative. We understand Cloud as resources pool and above there is a middleware system that runs virtualization.

Another definition describes Cloud as three things: thin client, grid computing and utility computing. Grid links computers to large infrastructure. Utility computing is responsible for "pay as you go" process for cloud. That allows to pay only for consumed resources.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_11,

Very important idea is also "access on demand". On demand provisioning means that client defines parameters of interest and pays for exactly what is ordered. If his needs grow “on demand”, he may increase the resources by buying more storage or performance and the additional service will be added on the fly and without stopping working applications and virtual machines.

There are two main divisions of the clouds. The first is the distinction between public, private and hybrid cloud. Public Cloud shares resources via the Internet and charges for utilization. Public Cloud is very easy to scale. Private Cloud use internal network and gives security and surveillance. Hybrid gives centralization and standardization, increased security but also flexibility, thanks to possible access via VPN clients. There is also possibility to set separated machine dedicated to client but on the vendor site.

The second divides the models of Clouds on Saas, PaaS, IaaS and Haas. Software as a Service (SaaS) rely on software distribution. The application is stored and made available to users through the Internet. This eliminates the need for installing and running applications on the client side. SaaS shifts the responsibilities of management, updates, support from the consumer to the supplier. Platform as a Service (PaaS) is a service, which gives virtual environment for work. This service is primarily aimed at developers. Infrastructure as a Service (IaaS) delivers computer infrastructure as a service. Recently new models appear, such as business as a Service.

Benefits of managing in the Cloud are: reduced cost, increased storage and flexibility. Other advantages are the independence of the architecture and portability. Reduced cost because limitation of need to purchase a license (or licenses cheaper and better used), no need of installation and software administration. Possibility of scaling the system without changing the infrastructure. Do not purchase the equipment in case of an emergency temporary demand. Pay as you go - we do not pay for equipment when it is not fully exploited. It is also important that small business does not need to employ technical support.

We have to ask how Cloud Computing will affect software development? Is migration of existing mechanisms possible? How to program large-scale distributed systems? How to work with large disk arrays? From the economical point of view we have even more doubts. How to estimate total cost of ownership Cloud? Which business model choose? How to provide availability and continuity of service. And the biggest conserve touches data security issues. Are my files safe? Is there any law regulation for Cloud? And finally: Are we ready for Cloud?

The top research subjects are: existing infrastructure utilization, migration of existing systems to Cloud, connecting mobile devices to Cloud, business as a service, fear of huge data leakage, hybrid clouds. There is also huge interest in green computing, which is understand as low energy consumption and low heat emission.

3 Scheduling Algorithm for Virtualization Tasks

In this section, we present the optimization problem in the Cloud Computing architecture. This is a real problem that grow up at our University. We show its transformation to the particular packing problem and meta-heuristic methods engaged to find an appropriate problem solution.

The most authors of tasks scheduling algorithms for CC systems discussed in literature [2,5,6] use mathematical model without consideration of many problems with desktop virtualization. Our new model for tasks scheduling of VM deployment for business and educational purposes is presented here. Especially in heterogeneous systems this task is an NP-complete.

In recent years CC systems have been rapidly developed. Undeniably for this type of system it is necessary to design new algorithms for management including scheduling. One of the problem that needs analysis is scheduling in for desktop virtualization CC systems. Solution is not obvious, because that problem is NP complete. Especially in a heterogeneous system, such as CC it is not trivial. Most of the proposed solutions use meta heuristic algorithms.

We also introduce genetic algorithm [1] for task scheduling in CC. Genetic Algorithms use techniques inspired by natural evolution (i.e. crossover, mutation) and are widely used for optimization problems.

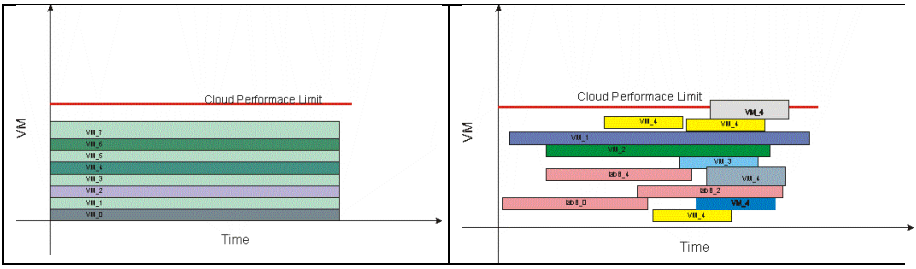


Fig. 1. Example workload from business and research applications

3.1 Problem Description

Methodology known from business problems do not fit to educational tasks. In light of this there is a need of novel design and analysis methodology for private Cloud for education. As it was mentioned, methods from industry do not apply to educational area (i.e. problem of performance parallelism).

Main difference is life cycle of virtual machine. Figure 1 shown two different life cycles of virtual machines. The first one can be found in companies like banks, where all employees work for eighth hours per day. The second is typical for research data centers, where clients tasks are not known a priori. None of those models fits academical environment. On next figure there is presented a example typical workload in educational systems.

In educational systems we may describe two goals. One is to deliver virtual machines for students labs. This machines are easy to predict, have typical configurations. In most cases low performance is required but inaccessibility is critical. Second is to deliver as much as possible other machines for research and teachers. This are hard to predict (in most case immediate deployment) and have various configurations but inaccessibility is not critical. Cloud System will face more problems like potential issues with untypical hardware (remote USB, GPU computing), temporary increasing of performance (parallel work in groups of students).

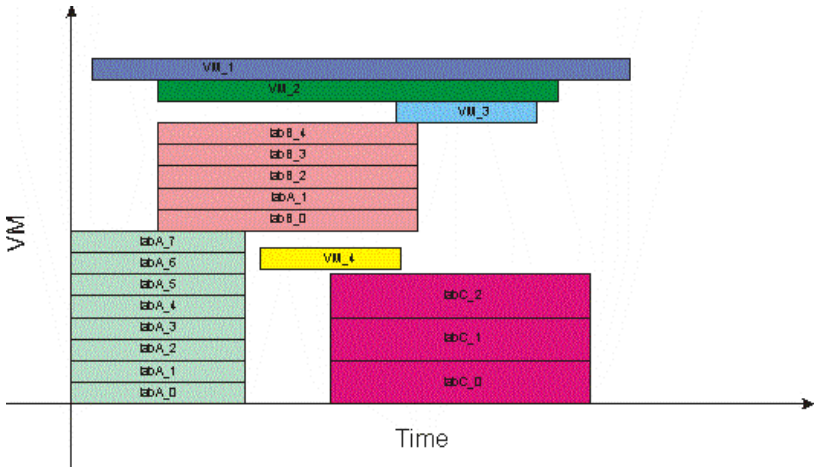


Fig. 2. VM Workload in educational tasks

We need new scheduling model because of untypical tasks, which are not compute intensive and do not have regular character. Tasks are also performance flexible, but not time flexible. Task can be cancelled by user but not delayed.

In university applications of planning tasks for the CC, there is a need to provide multiple virtual machines (VM) with different performance requirements, for the given time intervals. Academic Cloud is consisted of always available computers (host machines). Number of host machines and their specification are known. For some VM required working hours are known and are strictly given by university staff. Each of VM is also described by memory size and CPU performance. Users connect to Cloud to access virtual machines from terminals. In our model to schedule deployment of virtual machines we need information such as: the total number of machines and for each of them: minimum required CPU performance, minimum required memory size, working time. It is also necessary to place this tasks on a system time-line. Starting time can be precisely defined or it can be fluent. Furthermore, it is assumed that the hardware resources of at least one host machine is larger than those required by the virtual machine. It is also important to noticed, that in academic applications we have several other dependencies. For example server VM for classes has to be started before other VM start and can't be stopped when machines are still running. In considered system we have given number of virtual machines. Each of them is described by required resources (CPU performance, memory). It is assumed that these requirements may be different for each machine. On the other side there is a CC system consisting of multiprocessor computers (host machines) with defined parameters. It is assumed that each virtual machine must be deployed on a single host. It is not allow for a virtual machine to use resources from several hosts.

To solve this problem it is necessary to create its mathematical model. Model from [2] unfortunately, cannot be applied to the problem of tasks scheduling of desktop virtualization. Authors use the worst-case calculation time. In the considered problem, this approach does not apply because the deployment time (starting and end time) is

pre-set. Therefore, the key problem in the task of planning the distribution of virtual machines is to ensure the specified performance at a specified time (i.e. working hours, time classes for students). In light of this we cannot handle the worst-case calculation time.

In our model VM must be deployed on a single host. It is not allow for a virtual machine to use resources from several hosts. Host changes are prohibited when machine is running.

It is possible to run two and more than two virtual machines on one host. Host resources used by machine are reserved for it for whole machine's run time and can be used only by this machine. When virtual machine is stopped, resources reserved for it are immediately released and can be assigned to the next task.

Schedule algorithm deploys each virtual machine on selected host where it will be executed. Starting time is also specified in this process.

It is easy to notice that the problem described above is of the big complexity due to the great number of restrictions of different kinds, for the first. One may observe that our problem looks like a bin-packing problem whereas it is also similar from other points of view to the vehicle routing problem. Hosts may be treating like a truck (vehicles) and virtual machines are the objects to be carried. A special source of difficulty comes from the time relations between particular virtual machines. These make our problem similar to project management task and can make it unfeasible i.e. unable to be realized from time to time. These properties carry on the evident observation that it is impossible to write down our problem using only mathematical formulas. We propose new mathematical model for this problem:

k - the number of virtual machines
l - the number of blades in the host
$M = \{1,2,\dots,k\}$ set of virtual machines to me deployed
$B = \{1,2,\dots l\}$ set of virtualization hosts {blade servers}

For all virtual machines i from set M we define

c_i as required computational power (CPU) for VM i
m_i as required computational memory (RAM) form VM i

For all servers j from set B we define

b_j as available computational power (CPU) at server j
p_j as available computational memory (RAM) at server j

We also define moments of time when VM starts (set contains start time for all k machines)

$$S = (S_1, S_2, \dots, S_k) \tag{1}$$

and also stop time for all machines

$$C = (C_1, C_2, \dots, C_k) \tag{2}$$

In light of this we define also two matrixes $\theta_{t,k}$ and $\Delta_{k,l}$ where $t \in S \cup C$ denotes moment of time.

$$\theta_{t,k} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3}$$

Number 1 in matrix denotes that VM is running at time t

$$\Delta_{k,l} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4}$$

Number 1 in matrix $\Delta_{t,k}$ denotes that VM i is running on server j

We may also define constrains. First constrains guarantee that we do not exceed maximum computation power on blades.

$$\forall_t \forall_{j \epsilon B} \forall_{i \epsilon M} \sum \Delta_{i,j}\, \theta_{t,i} c_i < b_j \tag{5}$$

The second one guarantee that we do not exceed maximum available memory (RAM) on blades.

$$\forall_t \forall_{j \epsilon B} \forall_{i \epsilon M} \sum \Delta_{i,j}\, \theta_{t,i} m_i < m_j \tag{6}$$

Our optimization task (reduction of utilized blade servers) is defined as follows:

$$U(j) = \begin{cases} 0, if \ \sum_{i \in M} \Delta_{i,j} = 0 \\ 1, if \ \sum_{i \in M} \Delta_{i,j} \neq 0 \end{cases} \tag{7}$$

$$\sum_{j \in B} U(j) \rightarrow min \tag{8}$$

To obtain a useful tool for the problem solving we put into the motion meta-heuristic approach [4]. Namely, for our purposes we reworked an optimization procedures based on the idea of Genetic Algorithm and Harmony Search Algorithm. Both approaches need for the very beginning an appropriate coding procedures to conduct necessary operators easy. To check the feasibility of reworked solution we used same special projections from the space of genotypes to the space of phenotypes based on the Baldwin Effect. Between others, this idea made our algorithms more flexible for the future needs i.e. restrictions. One of most important task during planning is to measure real system parameters. We have introduced monitoring system for utilization of CPU, disk and network measurement in real laboratory.

Overview of our utilization analysis system for educational CC is presented on figure. We collect statistical data from our real laboratory network including processor load, number of processors, memory and storage usage. Than we prepare example workloads for CC system and we analyses scheduling algorithms and cloud configuration. It is also possible to do reconfiguration of analyzed system and recalculate nodes utilization. In table 1 we have details of 6 example workloads from our real laboratory system.

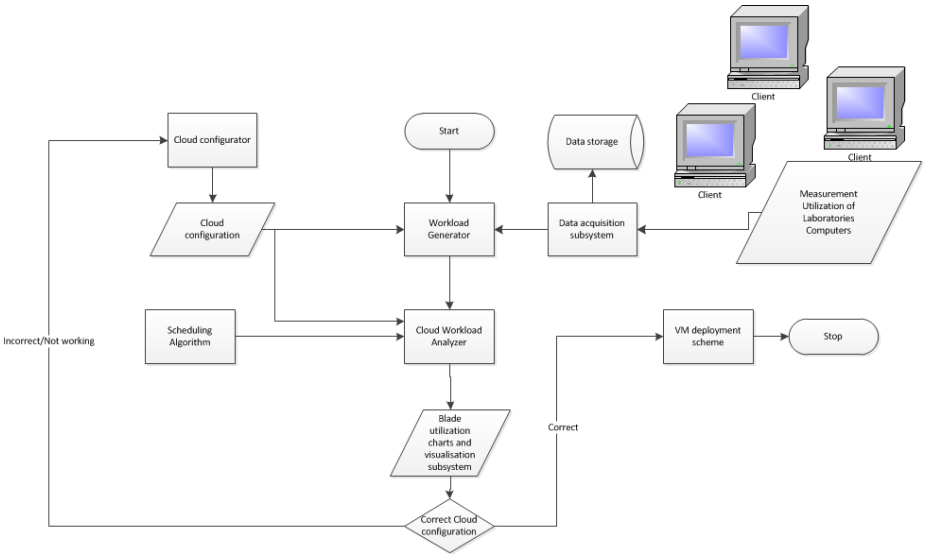


Fig. 3. Monitoring system for utilization of real laboratory

Table 1. Characteristics of workloads

Workload	Number of classes	Number of lab. Requested VM	Number of general VM (for teachers)	Total numer of VM
0	4	52	26	78
1	4	56	26	82
2	4	81	25	106
3	5	64	40	104
4	5	70	36	109
5	4	48	26	74

4 Conclusions

We consider virtualization as an interesting tool for education as well as for research purposes and also topic of our researches. In this paper we have presented mathematical model of virtualization for universities educational tasks. The presented work concerns the simulation of distributed systems, private clouds. The tool is to streamline the process of preparing the simulation, which is a first in-depth analysis of the structure of the test network and the tasks performed by it, and then at such a selection of the simulator and its parameters so that the obtained results best reflect the actual behavior of the tested system. This approach to solving the problem is very cost effective even for economic reasons.

Virtual machines deployment for educational purposes has other than business type restrictions. We have checked several deployment schemes and developed pre-scheduling reduces performance algorithm specially for students labs. Our future plans concern improvement of pre-scheduling and approximation of mathematical model coefficient from real system.

References

1. Greblicki, J., Kotowski, J.: Analysis of the properties of the harmony search algorithm carried out on the one dimensional binary knapsack problem. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2009. LNCS, vol. 5717, pp. 697–704. Springer, Heidelberg (2009)
2. Zhao, C., et al.: Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing. In: Proc. 5th Int. Conf. on Wireless Comm., Net. and Mobile Computing, WiCom 2009 (2009)
3. Voas, J., Bojanova, I., Zhang, J.: Cloud Computing. IT Professional 15(2), 0012–0014 (2013)
4. Kotowski, J.: The use of the method of illusion to optimizing the simple cutting stock problem. In: Proc. 7th IEEE Conference on Methods and Models in Automation and Robotics, MMAR 2001, vol. 1, pp. 149–154 (2001)
5. Garg, S.K., et al.: Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers. J. Parallel Distrib. Comput. (2010)
6. Cagan, J., Shimada, K., Yin, S.: A survey of computational approaches to three-dimensional layout problems. Computer Aided Design 34(8), 597–611 (2002)
7. Iosup, A.: Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. IEEE Transactions on Parallel and Distributed Systems 22(6) (2011)

Optimization and Control of Transport Processes in the Distributed Systems

Alexander Buslaev[1] and Mikhail Volkov[2]

[1] MADI, 64, Leningradskiy Prosp., Moscow, 125319, Russia
apal2006@yandex.ru
[2] MTUCI, 8a, Aviamotornaya street, Moscow, 111024, Russia
mmvolkov@gmail.com

Abstract. Algorithms for the control of movements of particles in a complex network were developed and implemented. The system performs monitoring of the traffic situation in metropolitan areas, may be demanded by special services, fire, police and emergency. The system uses proprietary algorithms of particle distribution and the creation of applications for terminal devices based on Android. It has the ability to collect data based boards Arduino.

Keywords: traffic, distributed systems, Arduino, distributed control.

1 Introduction

The control system of movement of particles is a complex network may be required in many spheres of human activity. Thus, similar systems can be demanded by special services, police, fire protection and emergency while patrolling of areas. The perspective directions of development and usage of the system is its application in medicine and logistics (access to goods in warehouses, large stores)[1].

The system optimizes movement of particles serving areas so that each point in the set area could be reached by one of the particles for a guaranteed time. Fire or police trucks, mobile research laboratories, automated loaders for transportation of freights or robots, including nanorobots (for medicine) can be presented as particles. Researches in the field of creation of the medical robots serving blood system are perspective in the world science [2].

Depending on application the scale of system and the graph changes. From kilometers (street road network) to meters (service of warehouse) and nanometers (medical nanorobots).

In existing analogs of system dynamic evolution of routes and shift of crews are not realized - route planning is carried out in advance and corrected from the control room [3]. Automatic scheduling capabilities are not provided.

1.1 Particles' Carrier – The Graph

The configuration of the graph is stored in the program in the form of a set of the vertices having geographical coordinates (width, longitude) connected by the edges

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_12,

having capacity (quantity of strips, stream speed at the moment) and a grid defining a relief. Thus, by imposing coordinates of any vertex on a relief grid it is possible to receive its coordinates in the three-dimensional space.
Particles are mobile objects that need to access all set of points on the graph in a minimum possible time T.

For each edge of the graph the speed of particles depends on the bandwidth of an edge and its workload. In case of the graph of a road network bandwidth depends on the quality of covering, quantity of strips and the existing restriction of speed on an edge [4].

2 Setting Objectives

2.1 Minimizing Access Time with the Resources Available

Distributed resource is a quantity of particles *n* serving a network.
Task 1 - distribution of available resources to minimize the access time of particles to the points on the graph without loading. A configuration of a network and quantity of particles of *n* is set in a task. It is necessary to find such an arrangement of particles which will allow to reach each point on the graph for a minimum time. In each time-point coordinating of particles has to be such to reach any set point for a guaranteed time (time depends on quality of a network). In a network without loading it is enough to consider weight of edges of the graph as their lengths.
Task 2 - minimization of access time of particles to points on the graph with permanent loading. In a network with permanent loading, for determination of weight of an edge in algorithms of routing on the column it is necessary to calculate its weight as edge's length on average speed on an edge.
Task 3 - minimization of access time of particles to points on the graph with dynamically changing loading. In a network with dynamically changing loading it is necessary to recalculate routes at each change of loading. The weight of edges of the count is calculated in the same way as in a case with permanent loading.

2.2 Calculation of the Minimum Number of Particles

Determination of a quantity of *n* and resource arrangement for the purpose of providing set time of an access *T*.
Task 4 - determination of a minimum quantity of particles of *n* necessary for a full covering of a free network. The configuration of a network and guaranteed time of access *T* for which particles have to reach any point of the graph is set. It is necessary to define the quantity of particles of *n* which is enough for covering of a network so that each point on the graph will be available for a guaranteed time *T*.
Task 5 - searching of a quantity and a configuration of particles on the loaded graph. In a network with permanent loading, for determination of weight of an edge in algorithms of routing on the graph it is necessary to calculate its weight as edge's length on average speed on an edge.

Task 6 - search of a quantity and a configuration of particles on the graph with dynamically changing loading. In a network with dynamically changing loading it is necessary to recalculate routes at each change of loading. The weight of edges of the graph is calculated in the same way as in a case with permanent loading.

3 Mesh Dividing the Area into the Field of Access

For distribution of crews the area taken by a road network on which search of an optimum arrangement of crews is carried out, needs to be broken into cells, in the amount equal to the number of crews.

For splitting area into cells templates as squares, hexagons, triangles are used. When covering a given grid area can be either set the cell size or number of cells. For a given cell size access to all points lying inside the cell can be guaranteed in a minimum time.

3.1 The Particle Distribution Network with Uniform Loading

For arrangement of three particles on the graph with uniform availability of vertices, the graph is divided into n of parts, where by n - quantity of particles. Each particle should have an access to m/n to vertices, where m - total of particles in the graph. The area in which the particle will be established is defined by area which is occupied by the chosen vertices.

Comparison of efficiency of splitting areas was executed. At the same size of a cell for a full covering of set road network it was required: squares - 16, triangles - 20, hexagons - 15.

Thus, the most effective is use of a hexagonal cell.

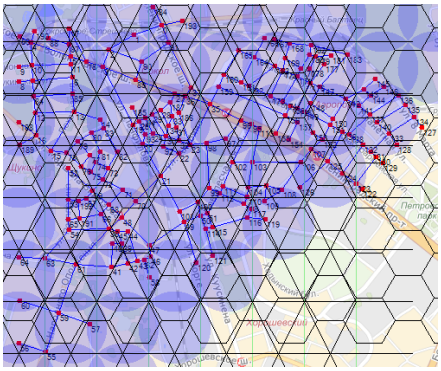


Fig. 1. Partition terrain options

For search of a configuration of an arrangement of particles on the column it is necessary to find border of area of the graph so that it has a minimum area. As border of

area the minimum convex cover round all vertices of the graph will be accepted. At an arrangement of particles they should not been settled down out of this cover.

3.2 The Concept of a Minimum Convex Hull

Let on the plane the final set of points (vertices of the graph) be set by *A*. Any closed line *H* without self-crossings such is called as a cover of this set that all points from *A* lie in this curve. If the curve of *H* is convex (for example, any tangent to this curve does not cross it more in one point), the corresponding cover is also called convex. At last, the convex cover of the minimum length (the minimum perimeter) is called as the minimum convex cover (hereinafter – MCC). All entered concepts are illustrated by the following drawing.

The main feature of MCC of a set of points of A is that these cover represents the convex polygon which vertices are some points of A. Therefore the problem of search of MCC finally is reduced to selection and ordering of the relevant points from A. Ordering is necessary for the reason that the polygon has to be an exit of an algorithm, i.e. a sequence of vertices.

Lets impose additional conditions about the vertices - direction of traversal of the polygon must be positive (recall that the positive figures called bypassing counter-clockwise). The task of creation of MCC is considered one of the simplest problems of computing geometry; there are many various algorithms for it. We will consider two such algorithms – Graham (Graham scan) [5] and Jarvis (Jarvis march) [6].

Listing 1: Comparison points

```
def rotate(A,B,C):
  return (B[0]-A[0])*(C[1]-B[1])-(B[1]-A[1])*(C[0]-B[0])
```

3.3 Jarvis's Algorithm

Jarvis's algorithm (other name - gift-wrapping algorithm) is arranged conceptually simpler than Graham's algorithm. It has two steps and does not demand sorting. The first step is exactly the same – the starting point which with guarantee enters MCC is necessary, the most left point from A will be taken [5].

The second step of MCC algorithm is under construction. Idea: to make the starting vertex as current, to look for the most right point in *A* relative to the current vertex, to make it current, etc. The process comes to an end when flowing again there will be a starting vertex. As soon as the point has got to MCC, it can be not considered further. Therefore we will get one more list *H* in which in the correct order MCC vertices will be stored. At once we enter starting vertex in this list, and in the list *P* we transfer this vertex to the end (where eventually we will find it and we will finish algorithm).

Now we will organize an infinite cycle on which each iteration we look for the most left point from *P* relatively the last vertex in *H*.

If this vertex starting, we interrupt a cycle if is not present - that we transfer the found vertex from *P* to *H*. After cycle end in *H* there is a required cover which we return as a result.

Listing 2: Complete code of Jarvis's algorithm

```
def jarvismarch(A):
  n = len(A)
  P = range(n)
  # start point
  for i in range(1,n):
    if A[P[i]][0]<A[P[0]][0]:
      P[i], P[0] = P[0], P[i]
  H = [P[0]]
  del P[0]
  P.append(H[0])
  while True:
    right = 0
    for i in range(1,len(P)):
      if rotate(A[H[-1]],A[P[right]],A[P[i]])<0:
        right = i
    if P[right]==H[0]:
      break
    else:
      H.append(P[right])
      del P[right]
  return H
```

We estimate the complexity of the Jarvis's algorithm. The first step is linear in n. In addition, in the second there is a nested loop, the number of outer iterations is the number of vertices in MCC h, the number of inner iterations does not exceed n. Consequently, the complexity of the entire algorithm is O (hn).

Unusual in this formula is that not only the length of the input data, but also the length of the output (output-sensitive algorithm) determines the complexity. A then depends on how the points will fall. In the worst case, all the points of A are included in the MCC (ie A is itself a convex polygon), then h = n and complexity jumps to quadratic. In the best case (with the proviso that from the point A do not lie on one straight line) $h = 3$ becomes linear and complexity. It remains to understand in advance what kind of case that can only be understood on the basis of the nature of the problem - if a lot of points and they uniformly fill a certain region, then (maybe) Jarvis's algorithm will be faster if the data is collected at the border area, the faster than the Graham's algorithm [6].

4 General Management Principles for the Distribution of Particles

T - the time during which the control object can be guaranteed up to a given target
V - speed based on network bandwidth
R - Radius crew area which can cater for the time T
ρ - flux density on the edges of the network
q - Intensity flows on the edges of the network

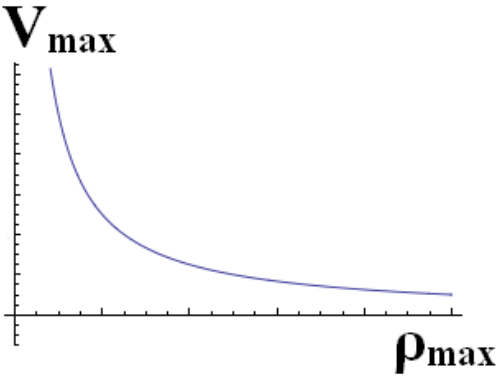


Fig. 2. The velocity of the flux density

Size field, any point of which can reach the particle.

$$Tv = R$$

For weighted graphs flux density on the edges of the network significantly

greater than zero

$$q >> 0$$

The particle velocity along the edge of the actual load is determined on the edge

$$v = V(q)$$

Flux density on the loaded column depends on the intensity of flows as the product of the intensity of the flow on the edge on the velocity of the particle

$$\rho(q) = q \cdot V(q)$$

Particle velocity at the loaded edge is defined as the ratio of the density to the intensity

$$V(q) = \frac{\rho(q)}{q}$$

4.1 The Control Algorithms to Minimize Access Time

In the algorithms to calculate the distance between the particles in a straight line, without regard to the topology of the graph. This assumption is possible for a network with uniform coating when the entire area is developed the same access network.[8] Control algorithms are chosen depending on the distance between objects. When you configure the algorithm parameters are chosen rules will work: a, b.

The first rule is valid when the two crews at a distance of less than a meter. When this situation arises, the crew with the lowest number is given the command "stop". The second crew continues to move.

The second rule is valid when the two crews at a distance of more than b meters. When this situation arises, the crew that should give the command "start a movement."

The third rule applies if in a neighborhood of c are several crews. In such a situation, it is necessary to choose a route that will lead to a uniform distribution of crews on the map. To find such a route, you must implement the possibility of calculating the scalar product of vectors derived from the edges of the graph, which are the crews, and the ribs, which may continue driving. Being in the vicinity of several crews R1, the rule applies to them 2.1, when the few crews in the neighborhood R2 2.2 rule applies when finding several crews in the vicinity of R3 applied to them, rule 1, if in the neighborhood of R3 there is only one crew, it can continue to move in any direction.

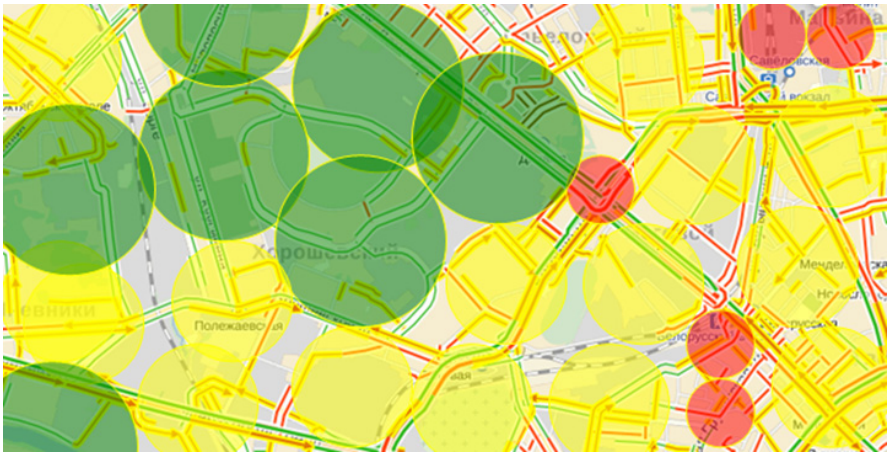


Fig. 3. The range of available

Rule 1. When approaching any of the crew to a fork in the system finds the edges, which are the other crews for these edges of vertex coordinates and direction of movement of the vehicles computed vector B1–n, where n - number of crews minus 1.

4 - crew for which will be chosen direction of movement
1,2,3 - crews, whose position will be taken into account when choosing a direction.
Calculated vector C_{1-n}, where n is the number of edges from the fork, for each edge of the graph coming out of the node to which the crew approaching. For each vector, retain the array coordinates of its start and end. The direction vector is determined using the point at which the crew was at the previous moment.
Calculates the dot product $Ai = B_1 - nC_i$.
In any Euclidean space (of dimension n) we can always choose an orthonormal basis $e_1, e_2, \ldots, e_n$ during the decomposition of vectors according to which:

$$a = a_1e_1 + a_2e_2 + \ldots + a_ne_n,$$

$b = b_1e_1 + b_2e_2 + \ldots + b_ne_n$

scalar product will be expressed by the following formula :

$$[a, b] = aTb = a_1b_1 + a_2b_2 + \ldots + a_nb_n \quad (1)$$

For two-dimensional space (the space GPS - coordinate plane can be
Considered from the land, because of its small size) the scalar product of two vectors, inner product can be calculated using the following formula :

$$(Lat_1t - Lat_1s) * (Lat_2t - Lat_2s) + (Lon_1t - Lon_1s) * (Lon_2t - Lon_2s) \quad (2)$$

where Lat_1t - latitude end of the first vector,
Lon_1t - longitude of the end of the first vector,
Lat_1s - the latitude of the first vector,
Lon_1s - longitude of the start of the first vector,
Lat_2t - latitude end of the second vector,
Lon_2t - longitude of the end of the second vector,
Lat_2s - the latitude of the second vector,
Lon_2s - longitude of the start of the second vector.

As recommended by the direction of motion is transmitted to the crew, the direction for which the angle between the directions of motion is minimal, hence cosϕ - maximum, therefore the maximum inner product divided by the product of the lengths of the vectors.
The Pythagorean Theorem of the coordinates of its start and end calculates vector length.

$$|X| = \sqrt{(Lat_1t - Lat_1s)^2 + (Lon_1t - Lon_1s)^2} \quad (2)$$

where Lat_1t - latitude end of the vector,
Lon_1t - longitude end of the vector,
Lat_1s - the latitude of the vector
Lon_1s - longitude of the start of the vector.

Rule 2. If any two of the crew approach to a distance of less than a1, then one of the crew (selected using a random number generator) is stopped and the data transmitted with it are not included in the analysis. Distance between crews is calculated as follows

$a = \sqrt{(Lat_1 - Lat_2)^2 + (Lon_1 - Lon_2)^2}$ where Lat_1 - Latitude of the first crew,
Lon_1 - longitude of the first crew,
Lat_2 - Latitude second crew,
Lon_2 - longitude of the second crew.
If there are n crews need to calculate n(n – 1) lengths.
Every 10 seconds to check crew, who stopped and if the distance between the crew stopped and any other moving crew over a2, stationary, the crew transferred command to start moving.

5 Software Implementation

The software part consists of two parts - server and client for mobile device. Taking in the account that all calculations are realized on the server the system may be quickly

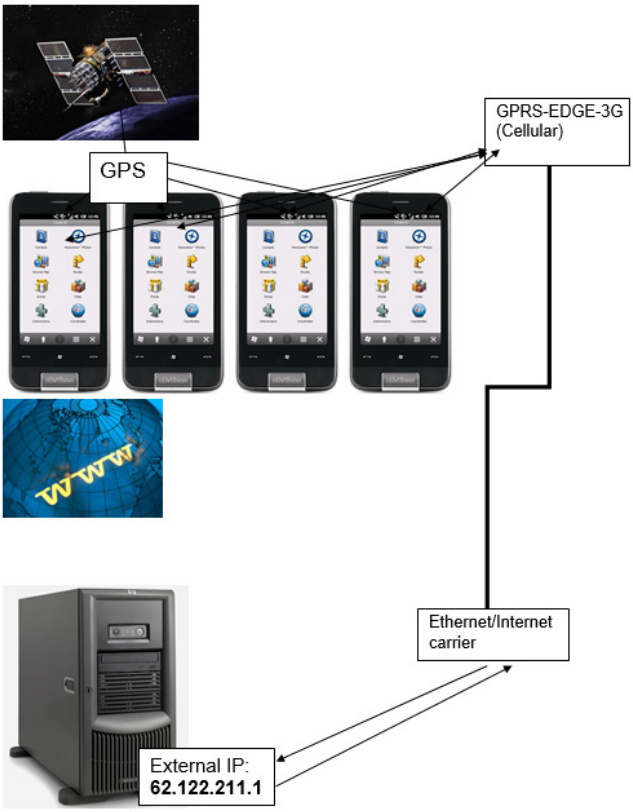


Fig. 4. System architecture

ported on various platforms: Android, iOS, WinPhone. The server part accepts data arriving from client devices, keeps statistics, records measurements needed to solve the experiment.

The client application queries the data collection boards for transmission of the measured data to the server and issue commands for operators of media client devices on where it is necessary to move for the objective performance.

6 Summary

The established system allows monitoring of distributed data collection intended to build the model in different fields for various needs. The system monitors the transport networks, pipelines and computer networks (LAN). Besides, it provides the possibility of a remote control. Introduction of such systems can significantly increase the effectiveness of the data's accuracy monitoring and, thus, accelerate the whole process of obtaining and processing of the data.

References

1. Buslaev, A.P., Tatashev, A.G., Yashina, M.V.: Stability of Flows on Networks. In: Proceedings of International Conference "Traffic and Granular Flows - 2005", pp. 427–435. Springer (2006)
2. Freitas Jr., R.A.: Current Status of Nanomedicine and Medical Nanorobotics. Journal of Computational and Theoretical Nanoscience 2, 1–25 (2005)
3. Buslaev, A.P., Tatashev, A.G., Yashina, M.V.: On Properties of the NODE System Connected with Cluster Traffic Model. In: International Conference of Applied Mathematics and Approximation Theory (AMAT) - 2012, Abstracts Book, pp. 66–67 (2012)
4. Buslaev, A.P., Tatashev, A.G., Yashina, M.V.: Stability of Flows on Networks. In: Proceedings of International Conference "Traffic and Granular Flows - 2005", pp. 427–435. Springer (2006)
5. Graham, R.L.: An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. Information Processing Letters 1, 132–133 (1972)
6. Jarvis, R.A.: On the identification of the convex hull of a finite set of points in the plane. Information Processing Letters 2, 18–21 (1973)
7. Buslaev, A.P., Gorodnichev, M.G.: Microwave Eye of "Big Brother": What is Visible from the Window of MADI. In: Ninth International Conference on Traffic and Granular Flow 2011. Book of abstracts. M. T - Comm, pp. 338–340 (2011)
8. Buslaev, A.P., Gorodnichev, M.G.: Microwave Eye of "Big Brother": What is Visible from the Window of MADI. In: Ninth International Conference on Traffic and Granular Flow 2011. Book of abstracts. M. T - Comm, pp. 338–340 (2011)

On Some Resources Placement Schemes in the 4-Dimensional Soft Degradable Hypercube Processors Network

Jan Chudzikiewicz and Zbigniew Zieliński

Military University of Technology, Faculty of Cybernetics,
ul. S. Kaliskiego 2, 00-908 Warszawa, Poland
{jchudzikiewicz,zzielinski}@wat.edu.pl

Abstract. In the paper some properties of the perfect resources placement in the 4-dimensional hypercube processors network with soft degradation are investigated. The two types of network processors (resource processors - I/O ports and working processors) are taken into consideration. In the work the notion of $(m, d|G)$ – perfect resources placement in a structure G of the hypercube type is extended to $(k|G)$ – perfect placement concept, that is a such allocation of k resources which minimizes the average distances between the working processors and resource processors in the structure G. Two algorithms for determining $(m, d|G)$ – perfect resources placement and the $(k|G)$ – resources perfect placement in a structure G of the hypercube network was given. The average number of working processors in the degraded 4-dimensional network with a given order (degree of degradation) for the $(1,1|G)$ resource placement is determined. This value characterizes the loss of the network computing capabilities resulting from the increase of the degree of network degradation.

Keywords: resources placement, hypercube network, fault-tolerant system.

1 Introduction

In a processors network, there may be resources, such as I/O processors or software components, that each processor needs to access. However, because of expense or frequency of use, resources in a network system might be shared to reduce the overall system cost. In general, then, the problem of resource placement is how should a limited number of copies of a resource be disseminated throughout a system giving comparable access to all processors. One of the situation to be considered in the context of resource allocation in fault-tolerant system is multiple-connection, in which every node without the resource is in connection with more than one copy of a resource. Multiple-connection gives rise to less contention, possibly yielding better performance, and reduces potential performance degradation after a copy is lost, because every node still can get access to one resource in the same number of hops.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_13,

One of considered in the literature the resource placement problem is a combination of the distance-d and the m adjacency problems, where a non-resource node must be a distance of at most d from m resource nodes [1]-[3]. In [4] a perfect deployment has been introduced and analyzed which is defined as the minimum number of resources processors which are accessible by working processors. Each working processor has access to the at least m resources processors at a distance of not more than d. The definition of perfect distance-d placement of m copies of resources in the processors network was fairly commonly used in torus-type networks [3],[4]. Several resource placement techniques have been proposed for the hypercube or cube-type network [5]- [7], and different error correcting codes have been used to solve this problem.

An interconnection network with the hypercube logical structure is a well-known interconnection model for multiprocessor systems [8]-[10]. Such networks possess already numerous applications in critical systems and still they are the field of interest of many theoretical studies concerning system level diagnostics ([11]-[13]) or resource placement problem [6], [14]. More commonly, we could observe the usage processors networks in critical application areas [15] (military, aerospace or medical systems etc.). Such networks are (mostly) used in real-time systems, which require a very high data reliability processing throughout all the network life cycle.

We investigate the 4-dimensional hypercube processors network with two types of processors. The first type of processors are resource processors (i.e. I/O ports). These processors mediate in communication with other systems (sensors networks) and keep data obtained from them. The second type of processors are working processors. These processors have to get access to data obtained from sensors and are processing data from sensors. We also assume that sensors are not mobile. The execution of some tasks by a working processor requires an access to resources, also some results obtained by working processors must be submitted by a resource processors to other systems [14].

Further we assume that 4-dimensional hypercube is a soft degradable processors network [11]. In such kind of networks a processor identified as faulty is not repaired (or replaced) but access to it is blocked. New (degraded) network continues work under the condition that it meets special requirements [14],[16], which may involve possibility of applying of some resources placement schema (e.g. processors with connected I/O ports). In the work [16] an analysis of the different patterns of reconfiguration in networks with soft degradation was conducted, where these schemes were divided into software and hardware ones. Analyzed in this paper the way of the hypercube network reconfiguration after identifying faulty processors is based on the determination of the coherent cyclic subgraph of the current structure with maximum cardinality [11]. If fault concern a resource processor an another I/O port allocation is required. Then I/O port may be switched to the determined fault-free processor.

A generalized cost of a network traffic with a specified resources deployment and workload of a network is usually tested with experimental methods or simulation. Distributing resources copies in a hypercube with an attempt to optimize system performance measures of interest has been investigated in [5].

The definition $(m,\ d)$ - perfect deployment is a characteristic of the value of the generalized cost of information traffic in the network at a given load of tasks. In the work [14] we have tried to apply the above mentioned approach to a processors network of a 4-dimensional cube type logical structure along with its soft degradation process.

In this paper we have extended the notion of $(m, d|G)$-perfect resources placement in a hypercube type structure G to $(k|G)$-perfect placement. It can be seen as a such allocation of k resources which minimizes the average distances between the working processors and resource processors in the structure G. Also, the algorithm for determining the $(k|G)$-*perfect placement* of resources in the network type hypercube was given.

The main aim of this paper is to apply and compare the above presented approaches to the 4-dimensional cube type processors network along with its soft degradation process. Particularly we were interested in obtaining the values of the average number of working processors for some $(m, d|G)$-resources perfect placement schemes depending on the degree of the network degradation.

The rest of paper consists of two sections and a summary. The second section provides a basic definitions and properties of working structures which are induced by the network in the process of its degradation. There is defined a concept of $(m, d|G)$-*perfect deployment* of processors with resources in the working structure of network G and the $(k|G)$-*perfect placement* . In the third section two algorithms for determining $(m, d|G)$-perfect placement and $(k|G)$-perfect placement are proposed. An illustration of the main algorithms steps for the same structure is given. Finally some concluding remarks are given.

2 The Basic Definitions and Properties

Definition 1. The logical structure of processors network we call the structure of 4-dimensional cube if is described by coherent ordinary graph $G = \langle E, U\rangle$ (E – set of processors, U – set of bidirectional data transmission lines), which nodes can be described (without repetitions) by 4-dimensional binary vectors (labels) in such a way that

$$\left[\delta\left(\varepsilon(e'),\ \varepsilon(e'')\right) = 1\right] \Leftrightarrow [(e',\ e'') \in U] \tag{1}$$

where $\delta\left(\varepsilon(e'),\ \varepsilon(e'')\right)$ is Hamming distance between the labels of nodes e' and e''.

The Hamming distance between two binary vectors $\varepsilon(\mathrm{e}')$ and $\varepsilon(\mathrm{e}'')$ complies with the dependency:

$$\delta\left(\varepsilon(e'), \varepsilon(e'')\right) = \sum_{k\in\{1,\ldots,n\}} (\varepsilon(e')_k \oplus \varepsilon(e'')_k)$$

where:

- $\varepsilon(\mathrm{e}')_{\mathrm{k}}$ – the k-th element of the binary vector $\varepsilon(\mathrm{e}')$,
- $\oplus$ – modulo 2 sum.

If $|E| = 2^4$ and $|U| = 2|E|$, then such graph we called (non labeled) 4-dimensional cube and denote by H^4.

Denoted by $\widetilde{H}^4 = \langle H^4; \{\varepsilon(e): e \in E(H^4)\}\rangle$ 4-dimensional cube labeled nodes, and by $\check{G}_p^{\&}(H^4)$ and $\check{G}_p^{\&}(\widetilde{H}^4)$ – a sets of coherent subgraphs of the graphs H^4 and $\widetilde{H}^4$ of order p, respectively, for class $\& \in \{C, A\}$ (C – cyclic graphs, A – acyclic graphs).

Example 1. The structure G (Fig. 1a) is a structure of type H^4 $\left(G \in \check{G}_{10}^{A}(H^4)\right)$ because the nodes of this structure can be labeled in accordance with the formula 1 (Fig. 1b).

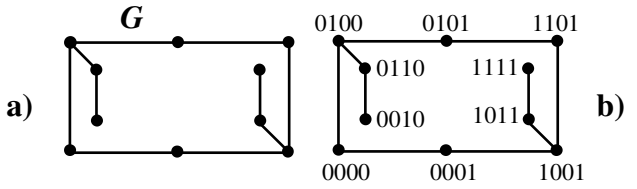


Fig. 1. Illustration of verification that the G structure is a structure of type H^4

The placement of resources in the processors network of 4-dimensional cube type logical structure we will regard as a labeled graph $\langle G; E\rangle$ where $G \in \check{G}_p(H^4)$ for $p \geq 6$ and $\dot{E} \subset E(G)$ ($\dot{E}$ – set of resource processors, $\{E(G) \setminus \dot{E}\}$ – set of the working processors of the network G).

Denoted by $d(e, e'|G)$ the distance between nodes e and e' in a coherent graph G, that is the length of the shortest chain (in the graph G) connecting node e with the node e'.

Definition 2. We say ([2],[4],[7]) that the labeled graph $\langle G; \dot{E}\rangle$ for $|\dot{E}| \geq 1$ is $(m, d|G)$ - ***perfect placement*** for $m \in \{1, \dots, \mu(G)\}, d \in \{1, \dots, D(G)\}, D(G)$ - diameter of the graph G if there exists the set $\dot{E}$ of minimum cardinality such that

$$\left[\forall_{e \in \{E(G) \setminus \dot{E}\}}: \left|\{e' \in \dot{E}: d(e, e'|G) \leq d\}\right| \geq m\right] \wedge \left[\forall_{\{e^*, e^{**}\} \subset \dot{E}}: d(e^*, e^{**}|G) > d\right] \wedge \left[(\mu(e''|G) = 1) \Rightarrow (e'' \notin \dot{E})\right].$$

Denoted by $E^{(d)}(e|G) = \{e' \in E(G): d(e, e'|G) \leq d\}$ for $d \in \{1, \dots, D(G)\}$ and $\hat{E}^{(1)}(G) = \{e \in E(G): (\exists_{e' \in E(e)}: \mu(e') = 1)\}$.

Denoted by $d_{max}(e, G) = \max\limits_{e' \in E(G)} d\big((e, e')|G\big)$, and by $E^{(d)}(e|G) = \{e' \in E(G): d(e, e'|G) = d\}$ for $d \in \{1, \dots, d_{max}(G)\}$, and by

$$\rho(e, G) = \left(\rho_1(e, G), \dots, \rho_{d_{max}(e,G)}(e, G)\right) \text{ for } \rho_d(e, G) = \left|E^{(d)}(e|G)\right|. \quad (2)$$

Definition 3. Denoted by $\varphi(e, G) = \sum_{e' \in E(G)} d(e, e'|G)$ for $e \in E(G)$ attainability of the processor e in the network G and by $\Phi(G) = \sum_{e \in E(G)} \varphi(e, G)$ attainability of the network G.

Using (2) we have

$$\varphi(e,G) = \sum_{d=1}^{d_{max(e,G)}} d\rho_d(e,G). \quad (3)$$

Example 2. We determined using (2) - $\rho(e,G)$, and using (3) - $\varphi(e,G)$, and $\Phi(G)$ for structures presented on the Fig. 2. We see that $d_{max(e,G_1)} = d(e_4, e_{10}|G_1) = 7$ and $d_{max(e,G_2)} = 4$. The determined value of $\rho(e,G)$, $\varphi(e,G)$ and $\Phi(G)$ are presented in table 1.

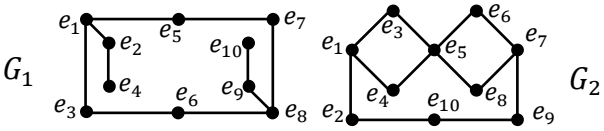


Fig. 2. Example of structures of type H^4 - G_1 $\left(G_1 \in \check{G}_{10}^{A}(H^4)\right)$ and G_2 $\left(G_2 \in \check{G}_{10}^{C}(H^4)\right)$

Table 1. The $\rho(e,G)$, $\varphi(e,G)$ and $\Phi(G)$ for the nodes of the structures G_1 and G_2 presented in the Fig. 2

	G_1								G_2				
$d(e,e'\|G)$ / $e \in E(G)$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	$\varphi(e,G_1)$	**1**	**2**	**3**	**4**	$\varphi(e,G_2)$
e_1	3	3	1	1	1	0	0	21	3	2	3	1	20
e_2	2	2	2	1	1	1	0	27	2	3	2	2	22
e_3	2	3	3	1	0	0	0	21	2	4	2	1	20
e_4	1	1	2	2	1	1	1	35	2	4	2	1	20
e_5	2	3	3	1	0	0	0	21	4	2	2	1	18
e_6	2	3	3	1	0	0	0	21	2	4	2	1	20
e_7	2	3	3	1	0	0	0	18	3	2	3	1	20
e_8	3	3	1	1	1	0	0	25	2	4	2	1	20
e_9	2	2	2	1	1	1	0	27	2	3	2	2	22
e_{10}	1	1	2	2	1	1	1	35	2	2	4	1	22
					$\Phi(G_1)$			251			$\Phi(G_2)$		206

Property 1. $\Phi(H^4) = 512$ because $\forall_{e \in E(H^4)}: (d_{max}(e,H^4) = 4 \wedge \rho_d(e,H^4) = \binom{4}{d})$. Using (3) we have $\forall_{e \in E(H^4)}: \varphi(e,H^4) = 32$ and $|E(H^4)| = 2^4$, then

$$\Phi(H^4) = |E(H^4)|\, \varphi(e,H^4).$$

Property 2. $d_{max}(e,G) = 8$ when the structure G is a Hamiltonian cycle in H^4.
Property 3. If $G' \subset G''$ that $\{E(G'')\backslash E(G')\} = e^*$, then $\forall_{e \in \{E(G'')\backslash\{e^*\}\}}: \varphi(e,G'') = \varphi(e,G') + d(e,e^*|G'')$.

Definition 4. Denoted by $\psi(e,f)=\sum_{e'\in\dot{E}(f)} d\left(e,e'\middle|G(f)\right)$ for $e\in\dot{E}(f)$ accessibility of the processor e in the placement f and by $\Psi(e,f)=\sum_{e\in\dot{E}(f)}\psi(e,f)$ accessibility of the resource processors in the placement f.

Denoted by $d(f)$ average distances between the working processors and resource processors in the placement $f\in F_k(G)$ for $1<k<\left|\left|E\left(G(f)\right)\right|/2\right|$. Then

$$d(f)=\left[k\left(\left|E\left(G(f)\right)\right|-k\right)\right]^{-1}\left(\sum_{e\in\{E(G(f))/\dot{E}(f)\}}\sum_{e'\in\dot{E}(G(f))} d\left(e,e'\middle|G(f)\right)\right) \quad (4)$$

where $k=\left|\dot{E}(f)\right|$.

Definition 5. Let $f\in F_k(G)$ be $(k|G)$ – ***perfect placement***, then

$$d(f)=min\{d(f'): f'\in F_k(G)\}. \quad (5)$$

Denoted by $\Omega(f)$ distances between the working processors and resource processors in the placement f. Let $\Omega_e^{\Delta}(f)$ denote increment $\Omega(f)$ when add $e\in\left\{E(G)\backslash\dot{E}(f)\right\}$ to the set $\dot{E}(f)$. Then

$$\Omega_e^{\Delta}(f)=\varphi\left(e,G(f)\right)-2\sum_{e'\in\dot{E}(f)} d\left(e,e'\middle|G(f)\right). \quad (6)$$

3 The Method and the Algorithm for Determining a Resources Placement

3.1 Determining $(\mathbf{m},\mathbf{d}|\mathbf{G})$ – Perfect Placement

For G as the first node we choose such a node $\mu(e_i)=\max_{e\in E(G)}\mu(e)$ that the subgraph $\bar{G}^{(d)}(G,e_i)=\langle\{E(G)\backslash E^{(d)}(e_i)\}\rangle_G$ has the smallest number of components of coherence. Then we determine a placement for every of these components of coherence, wherein if a component of coherence is one-node it belongs to the set $\dot{E}(f)$.

Based on the presented method was developed the algorithm (I) for determining $(m,d|G)$- ***perfect placement.***

Step 1.

Choose a node $e_i\in E(G)$ such that:

- the degree of $\mu(e_i)=\max_{e\in E(G)}\mu(e)$;
- subgraph $\bar{G}^{(d)}(G,e_i)$ has the smallest number of components of coherence.

Add the node e_i to the set $\dot{E}(f)$.

Step 2.

Check if a component of coherence of the subgraph $\bar{G}^{(d)}(G,e_i)$ is one-node or $\bar{G}^{(d)}(G,e_i)=\emptyset$.

YES

If $\bar{G}^{(d)}(G,e_i)\neq\emptyset$ add all nodes of $\bar{G}^{(d)}(G,e_i)$ to the set $\dot{E}(f)$.

Go to step 3.

NO

Assume that the $\bar{G}^{(d)}(G, e_i)$ is a new graph G.

Return to step 1.

Step 3.

The end of the algorithm (I).

An illustration of the algorithm work is presented in [14]. Algorithm will choose $(1,1|G)$ – perfect placement for the structure G_2 from Fig. 2. One of the possible $(1,1|G)$-perfect deployment for the structure G (chosen by the algorithm) is shown .

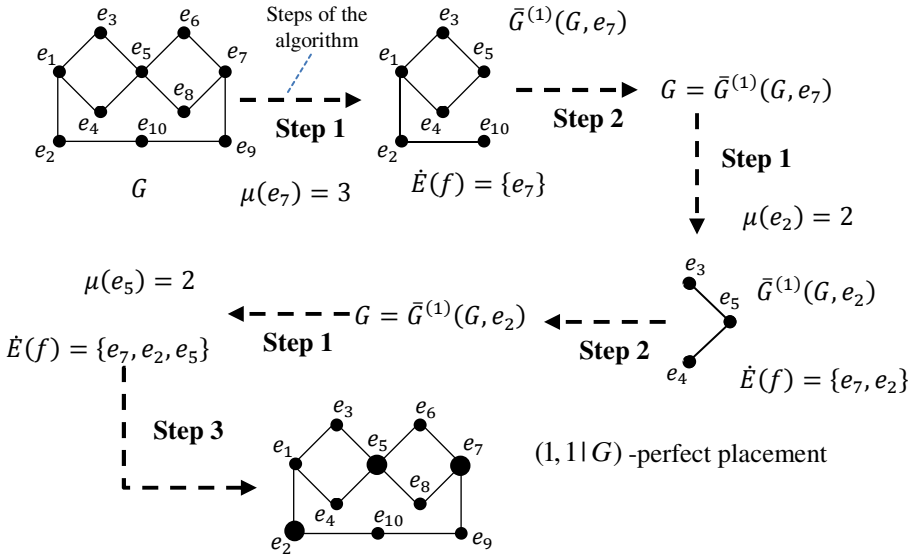


Fig. 3. An illustration of the algorithm (I) steps

3.2 Determining $(k|G)$ – Perfect Placement

The proposed method for determining $(k|G)$ – perfect placement is based on the calculation of $\Omega_e^{\Delta}(f)$: $\{E(G) / \dot{E}(f)\}$ (6). The method choose a node e_i such that $\varphi(e_i, G) = \min_{e \in E(G)} \varphi(e, G)$. The operations of calculation and selection of the node will be repeated for $k < \lfloor |E(G(f))|/2 \rfloor$. For every step is determined $(k+1|G)$ - perfect placement.

Based on the presented method was developed the algorithm (II) for determining $(k|G)$ - ***perfect placement.***

Step 1.

For the structure G using (3) determine $\varphi(e, G)$.

Step 2.

Choose a node $e_i \in E(G)$ such that $\varphi(e_i, G) = \min_{e \in E(G)} \varphi(e, G)$.

Add the node e_i to the set $\dot{E}(f)$.

Step 3.

Check if $k < \left|\left|E\left(G(f)\right)\right|/2\right|$.

YES

Using (6) determine $\Omega_e^{\Delta}(f)$: $\{E(G) \,/\, \dot{E}(f)\}$.

Return to step 2.

NO

Assume that the f is $(k|G)$ – ***perfect placement.***

Go to step 4.

Step 4.

The end of the algorithm.

An illustration of the algorithm work is presented in Fig. 4. Algorithm will choose $(k|G)$ – ***perfect placement*** for the structure G_2 from Fig. 2. The algorithm stop working for $k = 4$.

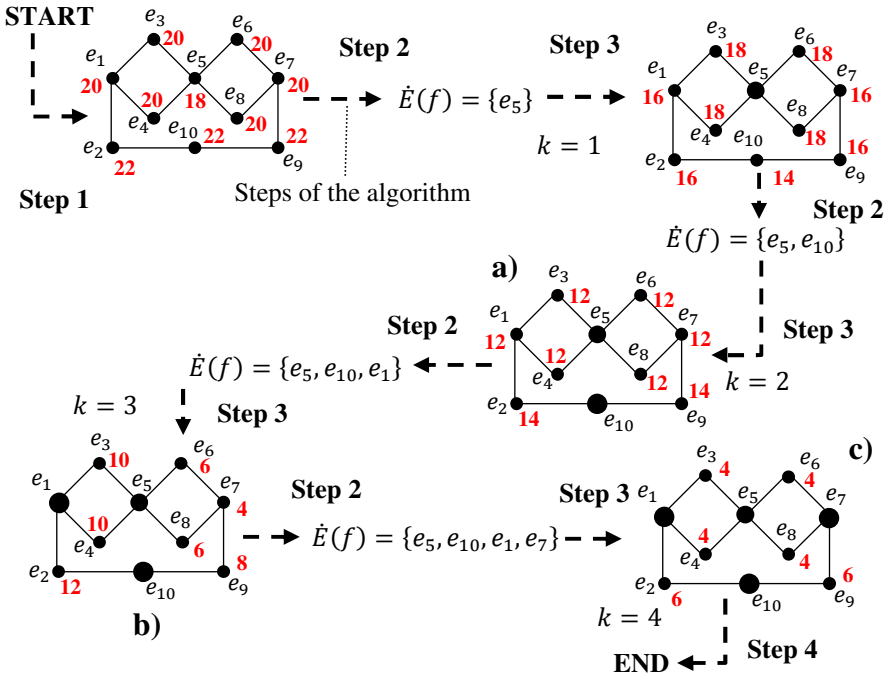


Fig. 4. An illustration of the algorithm (II) steps

The algorithm in 10 steps choose three placements for $k = \{2,3,4\}$ presented on the Fig. 3. a), b), c) respectively.

3.3 Remarks

Comparing the results of operation of both algorithms we can easily observe that the first one gives always $(1, d|G)$– perfect type placements (if such exists) while the second not always allow to obtain placement which is the $(1, d|G)$– perfect type

placements. For instance, the placement obtained in the Fig. 4c is $(4|G)$ – perfect placement and simultaneously is $(1,1|G)$ type placement but it is not the perfect placement because of the number of resources is not a set of minimal (is equal to 4 while the algorithm gives us perfect placement with 3 resource processors – Fig. 3).

The algorithm of determining the $(k|G)$ – perfect resources placement may be useful for obtaining approximate solutions for $(1,\delta|G)$– perfect type placements and $\delta\epsilon\{1,..,d_{max}\}$.

We have applied given in 3.2 algorithm (I) for determining the $(1,1|G)$-perfect placements in 62 cyclic working structures (i.e. within the 4-dimensional hypercube network structures with the degradation degree $0 \leq \varrho \leq 7$). By the degradation degree is meant the number equal to $\varrho = 2^4 - |E(G)|$). It was turned out that for some structures the $(1,1|G)$-perfect placements do not exist. Based on the known numbers of instances of these structures [11] in the 4-dimensional hypercube network the average numbers of working processors for the $(1,1|G)$-perfect placement depending on the degree of the network degradation were determined (Table 2) . This value characterizes the loss of the network computing capabilities resulting from the increase of the degree of network degradation.

Table 2. Characterization of the $(1,1|G)$- perfect placement for cyclic working structures in the 4-dimensional hypercube depending on the degree of the network degradation (ϱ)

	ϱ							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
% of existing placements	100	100	100	100	99	100	97	94
average number of working processors	12	11	10	9,19	8,46	8,17	6,92	6

4 Summary

In this paper the method of determining some schemes of resources placement in the 4-dimensional hypercube processors network with soft degradation was presented.

The $(m,d|G)$-perfect placement is a characteristic of a value of the generalized cost of information traffic in the network structure G at a given load of tasks. In the paper this notion was extended to the $(k|G)$-perfect placement which may be easily determined on the base of the accessibility measure of resources calculated as total sum of distances between the working processors and resources processors for the given placement in the working structure G. The algorithm of determining of $(k|G)$ -perfect placement was presented and it may be useful for obtaining approximate solutions for $(1,d|G)$ type perfect placements.

It should be noticed that real cost of information traffic in a network for a given deployment of resources processors depends on the nature of the tasks performed by

the network. We plan to examine this problem in the future with the use of simulation methods for a specified (m,d)-perfect deployments and a given type of task load of the network.

References

1. AlBdaiwia, B.F., Bose, B.: On resource placements in 3D tori. Journal of Parallel Distributed Computer 63, 838–845 (2003)
2. AlBdaiwia, B.F., Bose, B.: Quasi-perfect resource placements for two-dimensional toroidal networks. Journal of Parallel Distributed Computer 65, 815–831 (2005)
3. Bae, M.M., Bose, B.: Resource Placement in Torus-Based Networks. IEEE Transactions on Computers 46(10), 1083–1092 (1997)
4. Imani, N., Sarbazi-Azad, H., Zomaya, A.Y.: Resource placement in Cartesian product of networks. Journal of Parallel Distributed Computer 70, 481–495 (2010)
5. Tzeng, N.-F., Feng, G.-L.: Resource Allocation in Cube Network Systems Based on the Covering Radius. IEEE Transactions on Parallel and Distributed Systems 7(4), 328–342 (1996)
6. Chen, H., Tzeng, N.: Efficient resource placement in hypercubes using multiple-adjacency codes. IEEE Trans. Comput. 43(1), 23–33 (1994)
7. Moinzadeh, P., Sarbazi-Azad, H., Yazdani, N.: Resource Placement in Cube-Connected Cycles. In: The International Symposium on Parallel Architectures, Algorithms, and Networks, pp. 83–89. IEEE Computer Society (2008)
8. Chudzikiewicz, J.: Sieci komputerowe o strukturze logicznej typu hipersześcianu. Instytut Automatyki i Robotyki. Wojskowa Akademia Techniczna, Warsaw (2002) (in Polish)
9. Chudzikiewicz, J., Zieliński, Z.: Determining a non-collision data transfer paths in hypercube processors network. In: Embedded Systems - High Performance Systems, Applications and Projects, pp. 19–34. InTech, Rijeka (2012)
10. Chudzikiewicz, J., Zieliński, Z.: Reconfiguration of a processor cube-type network. Przegląd Elektrotechniczny (Electrical Review) 86(9), 149–153 (2010)
11. Zieliński, Z.: Podstawy diagnostyki systemowej sieci procesorów o łagodnej degradacji i strukturze hipersześcianu. Wojskowa Akademia Techniczna, Warsaw (2012) (in Polish)
12. Kulesza, R., Zieliński, Z.: The life period of the hypercube processors' network diagnosed with the use of the comparison method. In: Monographs of System Dependability - Technical Approach to Dependability, pp. 65–78. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2010)
13. Zieliński, Z., Strzelecki, Ł., Kulesza, R.: Diagnosability characterization of the 4 dimensional cube type soft degradable processors' network. In: Monographs on System Dependability – Problems of Dependability and Modeling, pp. 283–296. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2011)
14. Chudzikiewicz, J., Zieliński, Z.: Resources placement in the 4-dimensional fault-tolerant hypercube processors network. In: Proc. International Conference on Innovative Network Systems and Applications (iNetSApp), Kraków, Poland (2013)
15. Lubkowski, P., Laskowski, D.: The end-to-end rate adaptation application for real-time video monitoring. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 295–305. Springer, Heidelberg (2013)

16. Reddy, B.A.P.: Design, analysis, and simulation of I/O architectures for hypercube multiprocessors. IEEE Trans. Parallel Distributed Systems 1(2), 140–151 (1990)
17. Kulesza, R., Zieliński, Z.: Metoda generowania struktur logicznych sieci procesorów o łagodnej degradacji typu 4-wymiarowego sześcianu. Biuletyn WAT LX(4) (2011) (in Polish)
18. Byłak, M., Laskowski, D.: Assessment of network coding mechanism for the network protocol stack 802.15.4/6LoWPAN. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 75–82. Springer, Heidelberg (2013)

Efficient Training of Context-Dependent Neural Nets with Conjugate Gradient Algorithms

Piotr Ciskowski

Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology
Wybrzeże Wyspiańskiego 27, Wroclaw, Poland
piotr.ciskowski@pwr.edu.pl

Abstract. The paper addresses the problem of modeling highly nonlinear mappings of contextual nature with neural nets. The model of context-dependent multilayer neural net is recalled, along with basic training algorithms. Although the effectiveness of context-dependent nets has been proved theoretically, the lack of efficient and simple training algorithm implementations suppressed popularity of this neural net model.

In this paper efficient conjugate gradient training algorithms for context-dependent nets are developed and discussed, as well as illustrated with classification and regression problem.

1 Introduction

Neural nets are effective and statistically well-established algorithms for pattern recognition, regression and prediction. Problems solved by neural nets are often highly dimensional, with nonlinear dependencies between variables. It is natural to observe that some features supplied on net's inputs are crucial for the task, while other ones are less important, although not negligible. In other words, some features describe the problem directly, while other specify the context.

The idea of context-dependent training has been investigated by many authors, considering machine learning in general [8-9], as well as specifically the neural nets [10, 1, 7]. Two approaches which inspired the author to develop the model presented in this paper are presented in [10-11]. The former is a two-tier system for classifying the ECG signals into normal and ventricular beats, where the weights of the main neural net were multiplied by three second-order links of a patient model. The latter describes a hybrid neural net architecture which learns the problem of inverse Jacobian control of a robot arm. The nonlinear mapping between Cartesian velocity and joint coordinates is modeled by a linear neural net which weights, corresponding to the entries in the inverse Jacobian matrix, are learned by an auxiliary context network.

The context-dependent neural net model used in this paper assumes a division of net's inputs into primary and contextual. The former represent the problem being solved directly and correspond to the regular inputs of traditional nets, while the latter describe the context of the problem and are used to adjust the weights leading to

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_14,

primary inputs. Therefore, these weights are not constant values (after the training process is completed), but are functions of the context.

The paper describes the details of context-dependent neural net model and presents efficient conjugate gradient training algorithms for them. The effectiveness of the suggested algorithms is illustrated on two examples: 1) a synthetic classification task -- XOR problem for 8 points placed in two contexts with shifted decision boundaries, 2) real life regression task from *bodyfat* dataset. The clarity and simplicity of the suggested algorithms are emphasized, as well as the straightforward generalization of traditional perceptron and its training algorithms to the context-dependent model.

2 Context-Dependent Feedforward Neural Nets

The models of a context-dependent neuron and multi-layer feedforward net are presented in [3], while their properties, including the relation to traditional model, the Vapnik-Chervonenkis dimension, discriminating hypersurfaces and the use of contextual information, are discussed thoroughly in [2]. In this section the main concepts will be recalled as the foundation for the discussion of training algorithms.

2.1 The Model of a Context-Dependent Neuron

The context-dependent neuron is presented in fig. 1.

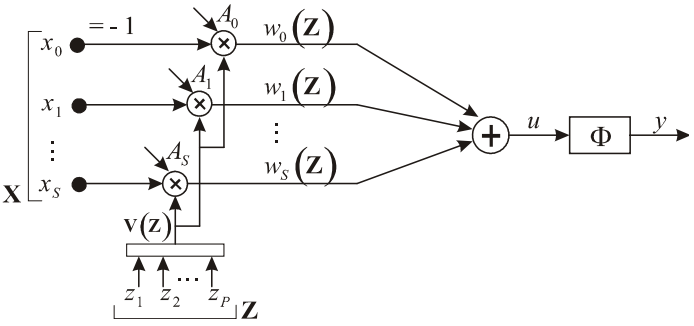


Fig. 1. The model of a context-dependent neuron

The main idea of this model is to separate primary and contextual inputs and operate on them in a different way. The neuron transforms primary inputs (vector **X**) similarly to the classical neuron, although the weights to these inputs are functionally dependent on the vector of contextual inputs $\mathbf{Z} = [\, z_1, z_2, \ldots, z_P \,]$. The continuous dependence of weights on the context vector is modeled with linear regression. The neuron's output is given by

$$y = \Phi\left[w_0(\mathbf{Z}) + \sum_{s=1}^{S} w_s(\mathbf{Z}) \cdot x_s \right] \tag{1}$$

where x_s is the s-th primary input, S is the number of primary inputs, while Φ is the activation function and w_0 is the neuron's bias.

The weights to primary inputs, and the bias, are modeled by the vector $\mathbf{V}(\mathbf{Z}) = \left[v_1(\mathbf{Z}), v_2(\mathbf{Z}), \ldots, v_M(\mathbf{Z})\right]^T$ of M independent basis functions of contextual inputs (basis functions are common for all weights in the net) and vectors of corresponding coefficients a_m, where $m = 1, 2, \ldots, M$. The weight w_s in the context specified by the vector $\mathbf{Z}$ is given by

$$w_s(\mathbf{Z}) = \sum_{m=1}^{M} a_{s,m} \cdot v_m(\mathbf{Z}) = \mathbf{A}_s^T \cdot \mathbf{V}(\mathbf{Z}) \tag{2}$$

where $\mathbf{A}_s$ is the coefficient vector $\mathbf{A}_s = \left[a_{s,1}, a_{s,2}, \ldots, a_{s,M}\right]^T$.

2.2 Context-Dependent Multi-layer Nets

The model of a context-dependent multi-layer net is presented in fig. 2. The vector of contextual inputs is used to adjust the weights in all layers. The difference between traditional and context-dependent model is that in the former the contextual inputs are added to the first layer's input vector and are directly used only by the first layer, while in hidden layers it is not possible to identify the influence of context data on the processing of primary inputs. In context-dependent net the context changes the whole mapping performed by the net - the character of the mapping is remained while its parameters change through the contexts.

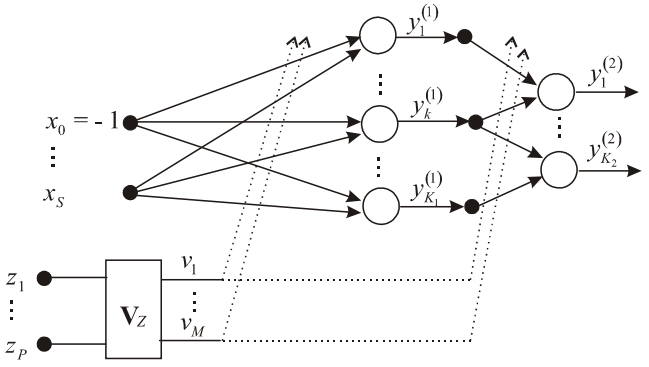


Fig. 2. Context-dependent multi-layer perceptron

As in traditional nets, we may group weight vectors of all neurons in the layer in the weight matrix $\mathbf{W} = \left[\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K\right]$, where K is the number of neurons in the layer, and then use it for calculating the output vector for the whole layer $\mathbf{Y} = \Phi\left(\mathbf{W}^T \cdot \mathbf{X}\right)$. As now the weights are functions of contextual inputs, calculation

of the layer's output should be preceded by calculation of weight values for current context. However, we suggest a one-stage algorithm:

$$y = \Phi\left[\mathbf{W}^T(\mathbf{Z})\cdot\mathbf{X}\right] = \Phi\left\{\mathbf{A}^T\cdot\left[\mathbf{X}\otimes\mathbf{V}(\mathbf{Z})\right]\right\} \tag{3}$$

where **A** is the matrix of coefficients, similar to the one used previously in (3).

Matrix **A** contains the coefficients for the weights of all neurons in the layer and is given by $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_K]$, where $\mathbf{A}_k$ is the coefficient vector for *k*-th neuron, concatenated from coefficient vectors for all its weights:

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{A}_{k,1} \\ \mathbf{A}_{k,2} \\ \vdots \\ \mathbf{A}_{k,S} \end{bmatrix} \tag{4}$$

3 Training Algorithms for Context-Dependent Nets

The special construction of coefficient matrix **A**, described in previous section, allows not only for faster and more direct output calculation, but is crucial for developing simple and efficient gradient-based training algorithms.

3.1 Gradient Based Training Algorithms for Context-Dependent Nets

Let us assume the error function of a context-dependent neuron, for one training example, in the form of the mean square error:

$$Q = \frac{1}{2}d^2 = \frac{1}{2}\left[y_d - y(\mathbf{X},\mathbf{Z})\right]^2 = \frac{1}{2}\left[y_d - \Phi\left(\mathbf{A}^T\left(\mathbf{X}\otimes\mathbf{V}(\mathbf{Z})\right)\right)\right]^2 \tag{5}$$

where y_d is the desired value on neuron's output. The error function's gradient with respect to the neuron's coefficient vector **A** is given by

$$\mathbf{g} = \nabla_{\mathbf{A}} Q = -d\cdot\Phi'(u)\cdot\left(\mathbf{X}\otimes\mathbf{V}(\mathbf{Z})\right) \tag{6}$$

All the elements correspond to the gradient of traditional neuron's error function with respect to its weight vector **W**, which would be given by: $\nabla_{\mathbf{W}} Q = -d\cdot\Phi'(u)\cdot\mathbf{X}$. The extension of the above formula to the case of multi-layer net with many outputs and a training set of many examples is straightforward.

Coefficient updates for the steepest descent training algorithm are given by

$$\mathbf{A}_{k+1} = \mathbf{A}_{k+1} - \eta_k\mathbf{g}_k \tag{7}$$

where $\mathbf{A}_k$ is the neuron's coefficient vector in k-th training step, η_k is the learning coefficient and the error function's gradient $\mathbf{g}_k$ in step k is given by (6).

As mentioned above, due to the special structure of matrix $\mathbf{A}$, the use of the Kronecker product, and the fact that the context-dependent neuron model is a generalization of the traditional one, all gradient-based training algorithms designed for traditional neural nets may easily be extended for context-dependent ones just by substituting the input vector $\mathbf{X}$ with the Kronecker product $\mathbf{X}\otimes\mathbf{V}(\mathbf{Z})$. The simple steepest descent algorithm may easily be supplemented with momentum component, or adaptive learning rate. More efficient conjugate gradient algorithms are presented in the next section.

3.2 Conjugate Gradient Algorithms for Context-Dependent Nets

Conjugate gradient algorithms for traditional neural nets assume that the updates to the vector $\mathbf{W}$ of all the weights in the net are applied in the direction which is conjugate to the search directions in all previous iterations.

Let us consider a vector $\mathbf{A}$ of all the context-dependent net's coefficients, used to model the dependence of each weight in the net on contextual variables. Let us generalize the traditional conjugate gradient training algorithm to context-dependent case.

Similarly to the traditional algorithm, in first iteration the updates of vector $\mathbf{A}$ should be equal to the negative gradient of the error function with respect to this vector. Therefore the first search direction is given by

$$\mathbf{p}_0 = -\mathbf{g} \tag{8}$$

where $\mathbf{g}$ is given by (6).

The next updates are applied according to

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \eta_k \cdot \mathbf{p}_k \tag{9}$$

where $\mathbf{p}_k$ is the current search direction

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1} \tag{10}$$

composed of the current gradient $\mathbf{g}_k$, computed according to (6), and the previous search direction $\mathbf{p}_{k-1}$. The constant β_k may be computed according to the Fletcher-Revees algorithm [5]:

$$\beta_k = \frac{\mathbf{g}_k^T \cdot \mathbf{g}_k}{\mathbf{g}_{k-1}^T \cdot \mathbf{g}_{k-1}} \tag{11}$$

or the Polak-Ribiere method [5]:

$$\beta_k = \frac{\Delta\mathbf{g}_{k-1}^T \cdot \mathbf{g}_k}{\mathbf{g}_{k-1}^T \cdot \mathbf{g}_{k-1}} \tag{12}$$

where in both cases the gradients are computed using (6).

The update size should be determined with a line search procedure, however a simplified method using adaptive learning rate also provides reasonable performance, as demonstrated later. The context-dependent version requires a periodical reset of coefficient updates to the negative gradient after some training time, similarly to the original method. The simple approach restarts when the number of training epochs reaches the number of net's parameters -- in this case it is the number of coefficients in vector **A**, which is equal to the number of net's weights multiplied by the number of base functions used to model the weights' dependence on the context. Another widely used method - the Powell-Beale algorithm, restarts when the orthogonality between the current and the previous direction is lost.

A short discussion on traditional and context-dependent nets' properties and training processes on two examples will accompany the classification application example presented in the next section.

3.3 Classification Example – XOR Problem in Several Contexts

The first synthetic example is an illustration of the idea of contextual classification. The task is to separate points into two classes as in standard XOR problem. The points are grouped in two contexts in such a way that the decision boundaries needed to separate them are rotated by 90 degrees (Fig. 3). Therefore the decision boundary has the same shape in both contexts and only changes its orientation from one context to another. The nature of the task is the same in both contexts, only with different parameters.

The problem is suitable for a 2-2-1 context-dependent net (2 inputs, 2 hidden neurons, one output) with 2 basis functions. In one fixed context it acts as a traditional 2-2-1 net . In traditional net, the information about the context of the point is just added to the standard input vector, increasing the input space by one. Contrary to that, in context-dependent net the contextual data is supplied on the contextual input, which is then transformed by the vector of basis functions, used to model the dependence of weights on the context. As shown in [2], in one fixed context the context dependent net has the same discriminating properties as a corresponding traditional net with the same number of weights, while increasing the basis function vector by one function adds another context, in which the decision boundaries of the net may be adjusted independently of the boundaries in other contexts.

The summary of training processes for 30 context-dependent 2-2-1 nets' are presented in Table 1. The nets were trained with different training algorithms, starting from the same initial coefficients for each algorithm. The following algorithms were used: *desc* - gradient descent, *desc-alr* - gradient descent with adaptive learning rate, *mom* - momentum, *mom-alr* - momentum with adaptive learning rate, *conj* - conjugate gradient algorithms using Fletcher-Reeves (*FR*) or Polak-Ribiere (*PR*) algorithm for calculating the β_k coefficient, and line search (*line*) or adaptive learning rate (*alr*) algorithm for determining the training step size.

All the nets used two basis functions and had unipolar sigmoid transfer functions in both layers. They were trained to maximum of 5000 epochs or until the mean square error was below 0.001. The most important indicator is the average number of epochs

to reach the desired error. As one can see, the conjugate gradient algorithms with line search outperform other algorithms. The results of a hybrid method (with adaptive learning rate algorithm for determining the step size) are also satisfactory, while less computationally demanding.

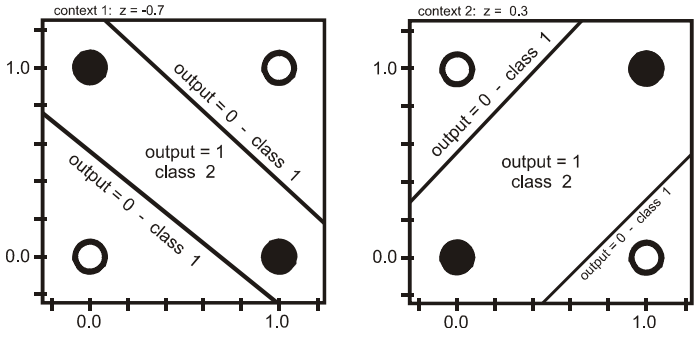


Fig. 3. XOR task in 2 contexts

For comparison, each context-dependent net was accompanied by a traditional 3-2-1 net, with the additional standard input supplied with point's context, initialized with the initial weights of context-dependent nets averaged in both contexts. Although traditional nets were also able to learn the task, the training time was much longer.

Table 1. Training results: XOR in 2 contents

	net	training method	MSE	epochs	epochs
			average		best
1	traditional	- mom-alr	0.00100	3439	2831
2	traditional	- conj-alr	0.00783	1752	913
3	context-dependent	- desc	0.00100	2768	1319
4	context-dependent	- desc-alr	0.00100	582	445
5	context-dependent	- mom	0.00100	2669	1328
6	context-dependent	- mom-alr	0.00573	887	439
7	context-dependent	- conj-FR-line	0.00077	130	69
8	context-dependent	- conj-PR-line	0.00082	267	48
9	context-dependent	- conj-FR-alr	0.00097	522	121
10	context-dependent	- conj-PR-alr	0.00100	799	366

The training record plot for a typical traditional net trained with momentum and adaptive learning rate is presented in fig. 4, while for context-dependent net trained with conjugate gradient algorithm with line search - in fig. 5. The plots present mean square error in both layers of the net and the whole net's classiffication error during training (the first layer's MSE error is estimated from backpropagation algorithm, while the second layer's error is real). Using conjugate gradient algorithm for training traditional net reduced training time, although did not smooth the error minimization

process as much as for context-dependent nets. This may be explained by the fact that traditional nets have to learn how to separate 8 points in 3-dimensional input space (including context), while context-dependent nets still learn to separate 4 points in 2-dimensional input space, while the dependence of separating boundaries on the context in on top of that process.

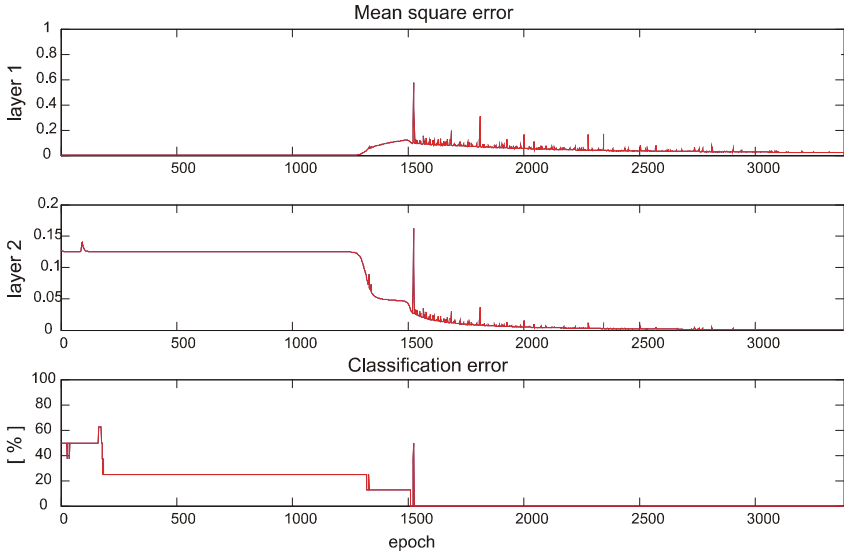


Fig. 4. Training example - classification problem: XOR in 2 contexts - traditional net training record

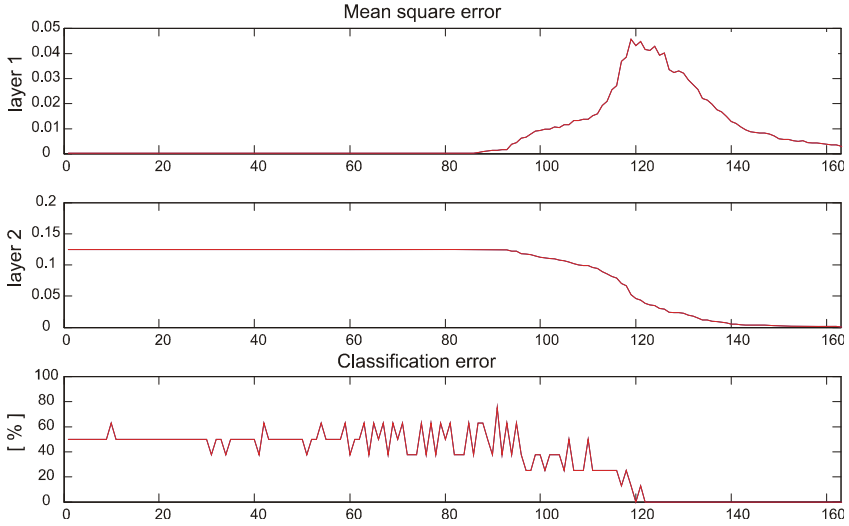


Fig. 5. XOR in 2 contexts - a context-dependent net training record

3.4 Regression Example - Bodyfat Dataset

The results of training context-dependent nets a regression problem are presented in Table 2. The widely known *bodyfat* dataset was used, made freely available by Dr. A. Garth Fisher with the publication of [6]. The task is to determine the percent of body fat based on 13 other patient's parameters. One of them was age, which was arbitrary chosen as the contextual input. Context-dependent nets with 5 hidden neurons and 3 basis functions were trained for 500 epochs. The average mean square error achieved with each training algorithm is given. Although the achieved error was not particularly impressive compared to the traditional net (what suggests a weak contextual dependency in this dataset, what is not analyzed in this paper), the gain over gradient descent algorithm may be seen. Let us also notice that the training time of context-dependent nets was only 2 times higher than for corresponding traditional net, while the number of adjustable parameters was 3 times higher for context-dependent nets. That advantage in computational complexity is a result of the construction of coefficient matrix and the use of on-stage output calculation. On the other hand, the observed longer time of conjugate gradient algorithms using line search may be minimized by parallelizing the error function calculations.

Table 2. Training results: bodyfat

	net	training method	MSE	time
1	traditional	- mom-alr	0.01176	68,77
3	context-dependent	- desc	0.01594	131,60
6	context-dependent	- mom-alr	0.01240	174,24
7	context-dependent	- conj-FR-line	0.00595	702,44
8	context-dependent	- conj-PR-line	0.00539	676,14
9	context-dependent	- conj-FR-alr	0.00924	132,91
10	context-dependent	- conj-PR-alr	0.00641	133,31

4 Conclusions

Context-dependent multi-layer feedforward neural nets are powerful tools for solving problems with complex nonlinear dependencies between variables.

In this paper efficient conjugate gradient training algorithms were presented and illustrated with classification and regression problems. The implementation of fast training algorithms will allow for application of context-dependent nets in more real life problems, including classification, regression and time-series analysis tasks.

A promising research field may be applying context-dependent nets in neural identification and control of complex systems, where the primary input-output mapping of the modeled object is dependent on some additional external variables. An example of such application was presented in [4]. In that paper a model context-dependent feedforward net was used to model a magnetorheological damper, which output – the force applied a building as a reaction to the measured displacement – is also dependent on its operational voltage. Context-dependent neural nets provide models, which are well fitted to

the contextual nature of such mappings, what results in faster training, better convergence and error using smaller size nets, compared to standard neural nets.

Further research will cover such problems as contextual version of the Levenberg-Marquardt training algorithm, advanced methods of contextual pruning, as well as developing new architectures, e.g. context-dependent RBF and recurrent nets.

References

1. Berthouze, L., Tijsseling, A.: A Neural Model for Context-Dependent Sequence Learning. Neural Processing Letters 23, 27–45 (2006)
2. Ciskowski, P.: Learning of Context-Dependent Neural Nets, Ph.D. thesis, Wroclaw University of Technology (July 2002)
3. Ciskowski, P., Rafajlowicz, E.: Context-Dependent Neural Nets - Structures and Learning. IEEE Transactions on Neural Networks 15(6), 1367–1377 (2004)
4. Ciskowski, P.: Contextual Modeling Using Context-Dependent Feedforward Neural Nets. In: Blackburn, P., Ghidini, C., Turner, R.M., Giunchiglia, F. (eds.) CONTEXT 2003. LNCS, vol. 2680, pp. 435–442. Springer, Heidelberg (2003)
5. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing, Boston (1996)
6. Penrose, K., Nelson, A., Fisher, A.: Generalized Body Composition Prediction Equation for Men Using Simple Measurement Techniques. Medicine and Science in Sports and Exercise 17(2), 189 (1985)
7. Arritt, R.P., Turner, R.M.: Context-Sensitive Weights for a Neural Network. In: Blackburn, P., Ghidini, C., Turner, R.M., Giunchiglia, F. (eds.) CONTEXT 2003. LNCS, vol. 2680, pp. 29–39. Springer, Heidelberg (2003)
8. Turney, P.: The Identification of Context-Sensitive Features: A Formal Definition of Context for Concept Learning. In: 13th International Conference on Machine Learning (ICML 1996), Workshop on Learning in Context-Sensitive Domains, Bari, Italy, pp. 53–59 (1996)
9. Turney, P.: The Management of Context-Sensitive Features: A Review of Strategies. In: 13th International Conference on Machine Learning (ICML 1996), Workshop on Learning in Context-Sensitive Domains, Bari, Italy, pp. 60–66 (1996)
10. Watrous, R., Towell, G.: A Patient-Adaptive Neural Network ECG Patient Monitoring Algorithm. In: Computers in Cardiology, Vienna, Austria, September 10-13, pp. 10–13 (1995)
11. Yeung, D.T., Bekey, G.A.: Using a Context-Sensitive Learning for Robot Arm Control. In: IEEE International Conference on Robotics and Automation, Scottsdale, Arizona, May 14-19, pp. 1441–1447 (1989)

Analysis of Mutation Operators for the Python Language

Anna Derezińska and Konrad Hałas

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
A.Derezinska@ii.pw.edu.pl,
halas.konrad@gmail.com

Abstract. A mutation introduced into a source code of a dynamically typed program can generate an incompetent mutant. Such a mutant manifests a type-related error that cannot be detected before the mutant execution. To avoid this problem, a program mutation can be provided at run-time, or incompetent mutants should be automatically detected and eliminated. We showed that the latter solution can effectively be applied providing selected mutation operators. This paper discusses mutation operators to be used for mutation testing of Python programs. Standard and object-oriented mutation operators were applied to the Python language. Python-related operators dealing with decorators and collection slices were proposed. The operators were implemented in MutPy, the tool for mutation testing of Python programs, and experimentally evaluated.

Keywords: mutation testing, mutation operators, Python, dynamically typed programming language.

1 Introduction

The Python programming language [1] belongs to the eight most popular programming languages over the last decade [2]. As a dynamically typed language it might be more sensitive to small mistakes of programmers and more difficult to be tested in comparison to strongly typed programming languages, such as C++, Java or C#. Nevertheless, the efficient testing of Python programs is indispensable. Therefore, we tried to apply the mutation testing approach to Python programs.

Mutation testing is recognized as a beneficial method for evaluating of a test case suite as well as creating of effective test cases [3]. The main idea originates from the fault injection approaches in which a fault is intentionally introduced and should be revealed by adequate tests. A code of a mutated program is slightly changed in comparison to the original code. If only one change is applied we speak about *first-order mutation*. Introduction of many changes to the same program is called *higher-order mutation*. A modified program, which is named a *mutant*, can be run against a set of test cases. If after running a test a mutant behavior differs from the original program, the mutant is said to be *killed* by the test. If the test suite did not reveal the change, either these test cases are insufficient and should be supplemented, or the mutant is *equivalent* one and no test could kill it.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_15,

The changes of programs are specified with *mutation operators*. *Standard* operators, also called *traditional* or *structural* ones, are easily applied in any general purpose strongly typed programming languages, e.g. arithmetic '+' is substituted by '-'. In object-oriented languages, several additional operators are used, which are devoted to object-oriented features. There are several studies on mutation operators concerning strongly typed languages, including structural languages [4,5] and object-oriented ones [6-10]. Applicability of specific language features to mutation testing operators was discussed in [11]. Analogies and differences among operators of different languages were shown in [12]. The question is whether these operators are applicable in a dynamically typed language, especially in Python. The Python language has also some specific aspects that should be tested, and therefore they could be considered by Python-related mutation operators.

The main obstacle of applying the mutation approach to a dynamically typed language is a fact that a mutant often cannot be verified before the run-time. The type relations in a programming statement and their correctness are not known before the mutant is called. A mutant can be syntactically correct but violets dynamic binding properties of the type system at runtime, hence is *incompetent* one. The general solution would be providing all mutations during the program execution. This approach to mutation of dynamically typed programming languages was considered by Botacci in [13]. The ideas were discussed on JavaSript examples, but were not put into a praxis.

In this paper, we followed a more practical approach. Selected mutation operators only sometimes generate incompetent mutants. This sort of cases can be automatically detected at run-time, and a mutant like this will be not counted to results of mutation testing. Experiment results showed, that the overhead cost concerning the elimination of such incompetent mutants is not very big for this set of mutation operators. Incompetent mutants can be handled by a mutation tool.

Based on mutation operators used in different programming languages (Fortran, C, Java, C#) and on Python programming features, a set of operators was determined that is applicable to Python programs. In this paper, we presented this set, basic principles of operator selection and detailed description of some Python-related operators. An exhaustive discussion about adaptation of all operators to the Python statements is omitted due to brevity reasons and can be found in [14].

Mutation operators discussed in this paper were implemented and experimentally examined using MutPy - a mutation testing tool for Python programs [15,16]. The set of operators in MutPy is so far the most comprehensive among all mutation tools for Python programs. It is the only tool supporting OO operators in Python. Other tools are either simple and not updated like Pester [17] or a proof of concept with 2-3 operators like PyMuTester [18] and Mutant [19]. The only exception is a recently developed Elcap [20] that introduces mutations at the abstract syntax tree level, similarly to MutPy. Elcap supports 8 structural mutation operators.

The rest of this paper is structured as follows. In the next Section we discuss mutation testing of dynamic languages as well as standard and OO operators that can be used in Python programs. Section 3 introduces mutation operators that were designed for selected features of the Python language. Experiments on mutation operators with Python programs are presented in Section 4, and Section 5 concludes the paper.

2 Adaptation of Mutation Operators to the Python Language

In strongly typed programming languages, a code replacement defined by a standard mutation operator can be verified by a static analysis. Types of data, variables and operators, as well as type consistency rules are known at the compilation time. Therefore, we can assure that a mutated program will be correctly compiled, and a mutation is not a source of type-related errors encountering at run-time.

The application of some object-oriented mutation operators is more complicated because they can depend on various conditions, e.g. other classes in an inheritance path [8,9]. However, while checking sufficient correctness conditions it is possible to avoid invalid code modifications.

2.1 Mutation of Dynamically Typed Programming Languages

Generation of mutants in a dynamically typed language can provide problems of type control [13]. Without a knowledge about types of variables, a mutation tool could introduce an invalid mutation, i.e. a mutant execution ends with a type related exception. This problem can be illustrated by a Python code example. The original program includes the following code, shown on the leftt hand side. If AOR (*Arithmetic Operator Replacement*) mutation operator is applied, the following mutant can be generated:

Example before mutation:

```
def add(x, y):
  return x + y
```

after mutation:

```
def add(x, y):
  return x - y
```

A mutant like this could be run with various parameters. If *add* is called with integer values, e.g. *add(2,2)*, the mutation will be valid. The mutant outputs value of *0*, whereas the original program ends with *4*. However, the function *add* could also be executed with string parameters. For example, calling of *add('a', 'b')* gives in the original program concatenation of those two strings - 'ab'. On the other hand, mutated version of the function cannot be executed because there is no string subtraction operation. After this kind of call, the *TypeError* exception is raised. Before the program execution, we could not determine types of variables *x* and *y*, therefore a part of mutants generated by AOR would be *incompetent*.

2.2 Mutation Operators from other Languages Applied to Python

The analysis of existing mutation operators was founded on operators designed and successfully applied in mutation testing tools for the following languages: Fortran (Mothra [4]), C (Proteum [5]), Java (MuJava [7]), and C# (CREAM [22]). There are many structural operators, but considering a different syntax of the Python language, the set of structural operators was preliminarily reduced. For example, there are no *goto* and *do-while* statements, hence C-like operators that change this sort of instructions are not applicable for Python programs.

Then, a systematic review of the operators was performed on an initial set of potential operators. The set consisted of 19 traditional operators and 36 object-oriented ones. Selecting an operator, we took into account its applicability to Python and a manifestation of a possible fault made by a program developer. Moreover, we try to avoid operators that can generate a substantial number of incompetent mutants. This assessment was based on a program static analysis and preliminary experiments performed using a previous version of the MutPy tool.

In general, operators that demand to add a new element to a mutant are risky ones. More safe are operators that delete or change a program element. This idea can be illustrated by an example. A statement `y = ~ x` can be mutated by omitting a bitwise negation operator. The outcome is a valid mutant `y = x`. The reverse mutation, i.e. adding '~' operator, would generate in most cases an incompetent mutant.

Other mutation operators that generate mostly incompetent mutants deal with substitution of variables, objects or constants, with other variables, objects or constants. If a type of a programming item to be substituted is not known, a number of potentially generated mutants will be very high; much higher than in a strongly typed language. On the other hand, the most of mutants like these would be incompetent. Therefore, such operators were not included into the final set of operators to be implemented.

The Python language supports basic object-oriented concepts, thus we also try to adopt object-oriented mutation operators used in Java or C#. However, while analyzing these operators, we can perceive that many of them are not suitable for Python. Several keywords are not used in the way as in Java or C#, for example *override, base, static* before a class field. Some constructions, common to strongly typed languages, like type casting or method overloading are also not used. Therefore, about 20 object-oriented mutation operators were excluded.

In the next step, we omitted object-oriented operators that could be defined in Python, but require type-based verification during application. For example, operator PRV (*Reference Assignment with other Compatible Type)* was omitted. Without type verification, these operators would mostly generate incompetent mutants.

Other operators refer to an optional usage of selected structures. For example, in Java and C# a keyword *this* can in many cases be used or not. The analogous keyword *self* is obligatory used in Python. Therefore, the operator JTI that deletes this kind of keyword is not applicable to Python.

After the analysis, remaining operators adapted for Python were implemented in the mutation tool. They are listed in Table 1, indicating an operator category: S - structural and OO - object-oriented.; and one operator falls in both categories.

New operators related to specific constructions of the Python language were also proposed (*Python-related* column). They are discussed in Section 3.

The last column indicates trial operators that were considered for Python programs, but finally did not include into the recommended set of mutation operators supported by the MutPy tool. According to the program analysis and experiments they provide many incompetent mutants and/or do not significantly contribute to the quality evaluation of tests or mutation-based test generation.

A detailed specification of all operators can be found in [14].

Table 1. Mutation operators for Python implemented in MutPy v 0.3

	Operator	Category	Python-related	Trial
AOD	Arithmetic Operator Deletion	S		
AOR	Arithmetic Operator Replacement	S		
ASR	Assignment Operator Replacement	S		
BCR	Break Continue Replacement	S		
COD	Conditional Operator Deletion	S		
COI	Conditional Operator Insertion	S		
CRP	Constant Replacement	S		
DDL	Decorator Deletion	S, OO	v	
EHD	Exception Handler Deletion	OO		
EXS	Exception Swallowing	OO		
IHD	Hiding Variable Deletion	OO		
IOD	Overriding Method Deletion	OO		
IOP	Overridden Method Calling Position Change	OO		
LCR	Logical Connector Replacement	S		
LOD	Logical Operator Deletion	S		
LOR	Logical Operator Replacement	S		
ROR	Relational Operator Replacement	S		
SCD	Super Calling Deletion	OO		
SCI	Super Calling Insertion	OO		
SIR	Slice Index Remove	S	v	
CDI	Classmethod Decorator Insertion	OO	v	v
OIL	One Iteration Loop	S		v
RIL	Reverse Iteration Loop	S	v	v
SDI	Staticmethod Decorator Insertion	OO	v	v
SDL	Statement Deletion	S		v
SVD	Self Variable Deletion	OO		v
ZIL	Zero Iteration Loop	S		v

3 Python-Related Mutation Operators

Various notions of the Python language [1], such as decorator, index slice, loop reversing, membership relation, exception handling, variable and method hiding, *self* variable were considered for mutation operators. New operators designed for some features, and selected after preliminary experiments are presented below. Other Python features were used in the scope of the adopted operators, e.g. the membership operator *in* can be negated by the COI operator (*Conditional Operator Insertion*) [14].

3.1 Decorators

Decorator is a function wrapper, which transforms input parameters and a return value of an original function. In Python, decorators are mostly used to add some behavior to a function without violating its original flow. A decorator is a callable object that takes a function as an argument and returns also a function as a call result. In a Python program, all decorators should be listed before a function definition and ought to be started with '@' sign.

The following example presents a simple decorator, which prints all positional arguments and a return value of a given function.

```
def print_args_and_result(func):
    def func_wrapper(*args):
        print('Arguments:' , args)
        result = func(*args)
        print('Result:', result)
        return result
    return func_wrapper
```

This decorator can be applied to a simple *add* function as follows:

```
@print_args_and_result
def add(x, y):
    return x + y
```

After calling the decorated function, e.g. *add(1,2)*, we obtain the following output:

```
Arguments: (1,2)
Result: 3
```

Three mutation operators dealing with the decorator notion were proposed: DDL *Decorator Deletion*, CDI *Classmethod Decorator Insertion*, and SDI *Staticmethod Decorator Insertion*.

The DDL operator deletes any decorator from a definition of a function or method.

Example before mutation:

```
1 class X:
2     @classmethod
3     def foo(cls):
```

after mutation:

```
class X:

    def foo(cls):
```

While using the @*classmethod* decorator, a method of a specialized class object is assigned as a first argument. If this decorator is deleted, the first argument is a default one. The method can be called from an object instance *x.method()*, where *x* is an instance, or using a class *X.method()*, where *X* is a class name. The latter case results in an incompetent mutant, but both cases can only be distinguished at run-time.

The DDL operator can also delete the @*staticmethod* decorator. The number of arguments accepted by the method without this decorator is incremented. The mutant will be valid if the modified method has a default argument that was not given in a method call. Otherwise, the mutant will be incompetent.

The DDL operator also deletes other decorators. If more than one decorator is assigned to an artifact all of them will be deleted.

The remaining two operators dealing with delegates were classified as trial ones. Both operators insert a decorator to a method definition, namely CDI inserts *@classmethod* and SDI *@staticmethod.* Operators inserting a new element into a program are usually a source of incompetent mutants, and the same occurred here.

3.2 Collection Slice

An element of a collection can be referred in Python programs using the access operator '[]' with an appropriate index (or a key in case of the *dict* type). It is also possible to use this operator for obtaining a subset of a collection, so called *slice*. The syntax of the '[]' operator usage is the following:

```
collection[start:end:step]
```

where *start* is the index of the first element of the slice, *end* the last element, and *step* a metric between two consecutive elements of a slice. Each of these three arguments is optional. The slice elements are taken from the beginning of a collection if *start* is missing, and will go the collection end if no *end* element is defined explicitly. The following examples are valid slices of collection *x*:

```
x[3:8:2]    x[3:8]      x[::2]    x[3:]
```

Mutation operator SIR *Slice Index Removal* deletes one argument of the slice definition*: start*, *end* or *step*. Each removal gives a syntactically correct slice.

Example before mutation: after mutation:

```
x[1:-1]                    x[:-1]
```

3.3 Loop Reversing

A loop in a Python program is realized by the *for* statement. Any loop can be iterated in the reverse order, which is completed by function *reversed(),* a built-in function of the Python language.

Mutation operator RIL *Reverse Iteration Loop* changes direction of a loop iteration. However, the preliminary evaluation of the operator indicates that many equivalent mutants can be generated by the operator.

Example before mutation: after mutation:

```
for x in y:                for x in reversed(y):
```

4 Experiments

The mutation operators listed in Table 1. were implemented in MutPy - a mutation testing tool for Python 3.x source code [15,16]. It applies mutation on the abstract

syntax tree (AST) level of a program. MutPy also supports higher order mutation and code coverage analysis [22].

In this section, results of the first order mutation testing in dependence on different mutation operators are presented. We carried out experiments on four Python programs [15,22]. The results are summarized in Table 2. The number of all mutants generated using a particular mutation operator is shown in the column *All*. In further columns, a distribution of this number is given, including the number of mutants that were killed by tests, killed by timeout, classified as incompetent mutants and remain not killed. Mutants are *killed by tests* when a test output differs from the output of the original program. Mutants are *killed by timeout* when the execution time exceeds a time limit calculated as approximately five times the original execution time.

Table 2. Mutation testing results

Operator	Mutant number						Mutation Score
	All	Killed by tests	Killed by timeout	Incompetent		Not killed	
AOD	38	25	1	0	0.0%	12	68.4%
AOR	740	441	5	117	15.8%	177	71.6%
ASR	82	67	4	3	3.7%	8	89.9%
BCR	14	7	4	0	0.0%	3	78.6%
COD	141	121	0	8	5.7%	12	91.0%
COI	601	511	5	39	6.5%	46	91.8%
CRP	2378	1435	3	47	2.0%	893	61.7%
DDL	16	0	0	7	43.8%	9	0.0%
EHD	37	21	0	9	24.3%	7	75.0%
EXS	55	36	0	4	7.3%	15	70.6%
IHD	27	11	0	0	0.0%	16	40.7%
IOD	47	25	0	6	12.8%	16	61.0%
IOP	4	0	0	0	0.0%	4	0.0%
LCR	61	36	0	15	24.6%	10	78.3%
LOD	7	7	0	0	0.0%	0	100.0%
LOR	136	121	1	0	0.0%	14	89.7%
ROR	512	423	4	5	1.0%	80	84.2%
SCD	4	1	0	0	0.0%	3	25.0%
SCI	43	7	0	8	18.6%	28	20.0%
SIR	85	61	4	0	0.0%	20	76.5%
Sum	5028	3356	31	268	5.3%	1373	71.2%

A mutant is classified as *incompetent* if the *TypeEror* exception is raised during running it with tests. The exception is detected by the mutation tool, and this mutant is not counted to the overall mutation score of the program. The whole time of the mutation process is increased due incompetent mutants, proportionally to their number.

However, as we can observe in Table 2, there were only 5.3% of incompetent mutants in comparison to all generated mutants. The percentage of incompetent mutants is different for various mutation operators. There are few operators for which about 15% - 25% of mutants was incompetent. A high number of incompetent mutants (43.8%) was also detected for the DDL operator. This is one of Python-related operators included into the final operator set, and it confirmed suspicions about possible many incompetent mutants. On the other hand, the handling of incompetent mutants is transparent to a user, and the overall overhead (few %) can be accepted.

The last column gives the *mutation score*. This measure reflects the ability of tests to kill the mutants. The mutation score is calculated as a sum of mutants killed by tests and killed by timeout divided by all generated and competent mutants. The exact calculation of mutation score should exclude *equivalent* mutants, i.e. mutants that have behavior equivalent to the original program and cannot be killed. In the current version of the MutPy tool, automatic classification of equivalent mutants is not supported. Therefore, the results can be treated as a lower bound of the mutation score, so-called *mutation score indicator*. However, it can be observed that CRP operator generated extremely many mutants. Many of them were not killed but also were not equivalent to the original program, as, for example, they change a string value to be displayed and this message content was typically not verified by tests.

5 Conclusion

The selection of discussed mutation operators were based on two general premises. The operator should be firstly useful and secondly effectively applicable in the mutation testing process of a dynamically typed language. The considered mutation process assumed application of operators into an intermediate form of a program before the run time, and an automatic detection of incompetent mutants.

All mutation operators presented in the paper were implemented in the mutation tool and experimentally verified. Experiments showed that only a few percent of generated mutants were incompetent, which gave an upper bound of the time overhead. This confirms the practical solution used in the MutPy tool providing a wide scope of various operators. Further experiments on mutating Python programs consider some techniques of mutation cost reduction and higher-order mutation [22].

An open question remains an extension of the mutation process with mutation operators used in strongly typed languages, but omitted due to the tendency of generating many incompetent mutants. If a mutant like this deals with a programming feature that could be easily verified with static analysis, then the operator is not worthwhile to be implemented. Otherwise, if such an operator is creating valid mutants apart from the incompetent ones, and these mutants could be used for verifying important language features, then it could be beneficial to create an additional environment for the run-time generation of this kind of mutants.

References

1. Python Programming Language, http://python.org
2. TIOBE Programming Community Index, http://www.tiobe.com (visited January 2014)
3. Jia, Y., Harman, M.: An Analysis and Survey of the Development of Mutation Testing. IEEE Transactions on Software Engineering 37(5), 649–678 (2011)
4. King, K.N., Offutt, A.J.: A Fortran Language System for Mutation-based Software Testing. Software - Practice and Experience 21(7), 685–718 (1991)
5. Delamaro, M.E., Maldonado, J.C.: Proteum-A tool for the Assessment of Test Adequacy for C Programs. In: Proc. of the Conf. on Performability in Computing Systems (PCS 1996), pp. 79–95 (1996)
6. Ma, Y.-S., Kwon, Y.-R., Offutt, A.J.: Inter-Class Mutation Operators for Java. In: Proc. of Inter. Symp. on Soft. Reliability Eng., ISSRE 2002, pp. 352–363. IEEE (2002)
7. Ma, Y.-S., Offutt, J., Kwon, Y.-R.: MuJava: an Automated Class Mutation System. Software Testing, Verification and Reliability 15(2), 97–133 (2005)
8. Derezińska, A.: Advanced Mutation Operators Applicable in C# Programs. In: Sacha, K. (ed.) Software Engineering Techniques: Design for Quality. IFIP, vol. 227, pp. 283–288. Springer, Boston (2006)
9. Derezińska, A.: Quality Assessment of Mutation Operators Dedicated for C# Programs. In: Proc. of 6th Inter. Conf. on Quality Software, QSIC 2006, pp. 227–234. IEEE Computer Soc. Press, Los Alamitos (2006)
10. Derezińska, A., Kowalski, K.: Object-Oriented Mutation Applied in Common Intermediate Language Programs Originated from C#. In: Proc. of 4th Inter. Conf. Software Testing Verification and Validation Workshops (ICSTW), pp. 342–350. IEEE Comp. Soc. (2011)
11. Derezińska, A.: Analysis of Emerging Features of C# Language Towards Mutation Testing. In: Mazurkiewicz, J., Sugier, J., Walkowiak, T., Zamojski, W. (eds.). Models and Methodology of System Development, Monographs of System Dependability vol. 1, pp. 47-59. Publishing House of Wrocław University of Technology, Wrocław (2010)
12. Boubeta-Puig, J., Medina-Bulo, I., Garcia-Dominguez, A.: Analogies and Differences between Mutation Operators for WS-BPEL 2.0 and Other Languages. In: Proc. of 4th Inter. Conf. on Soft. Testing, Verif. and Validation Workshops, pp. 398–407. IEEE (2011)
13. Bottaci, L.: Type Sensitive Application of Mutation Operators for Dynamically Typed Programs. In: Proc. of 3rd International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp. 126–131. IEEE Comp. Soc. (2010)
14. Derezińska, A., Hałas, K.: Operators for Mutation Testing of Python Programs. Res. Rep. 2014, Inst. of Comp. Science Warsaw Univ. of Technology (2014)
15. Hałas, K.: Cost Reduction of Mutation Testing Process in the MutPy Tool. Master thesis, Institute of Computer Science, Warsaw University of Technology (2013) (in Polish)
16. MutPy, https://bitbucket.org/khalas/mutpy
17. Jester - the JUnit test tester, http://jester.sourceforge.net/
18. PyMutester, http://miketeo.net/wp/index.php/projects/python-mutant-testing-pymutester
19. Mutant, http://github.com/mikejs/mutant
20. Elcap, http://github.com/sk-/elcap
21. Derezińska, A., Rudnik, M.: Quality Evaluation of Object-Oriented and Standard Mutation Operators Applied to C# Programs. In: Furia, C.A., Nanz, S. (eds.) TOOLS 2012. LNCS, vol. 7304, pp. 42–57. Springer, Heidelberg (2012)
22. Derezińska, A., Hałas, K.: Experimental Evaluation of Mutation Testing Approaches to Python Programs. In: Proc. of IEEE Inter. Conf. on Software Testing, Verification, and Validation Workshops. IEEE Comp. Soc. (in press, 2014)

Deterministic Schedule of Task in Multiprocessor Computer Systems with Higher Degree of Dependability

Mieczyslaw Drabowski[1] and Edward Wantuch[2]

[1] Cracow University of Technology Faculty of Electrical and Computer Engineering, Poland
drabowski@pk.edu.pl
[2] AGH University of Science and Technology
Faculty of Mechanical Engineering and Robotics, Poland
ewantuch@agh.edu.pl

Abstract. The paper includes a proposal of a new approach of coherent concurrent task scheduling and resource assignment, which are characteristic for the problem of dependable system synthesis. Task scheduling, resource partition, task and resource allocation are basic problems in high-level synthesis of computer systems. Synthesis may have a practical application in developing tools for computer aided rapid prototyping of such systems.

Keywords: synthesis of computer system, scheduling, multiprocessors, dependability.

1 Introduction

The goal of high-level synthesis of computer systems is to find an optimum solution satisfying the requirements and the constraints enforced by the given specification of the system. The following criteria of optimality are usually considered: costs of system implementation, its operating speed and dependability, i.e. reliability, availability and fault tolerance.

A specification describing a computer system may be provided as a set of interactive tasks. In any computer system certain tasks are implemented by hardware. The basic problem of system synthesis is partitioning system functions due to their hardware and software implementation. The goal of the resource assignment is to specify what hardware and software resources are needed for the implementation and to assign them to specific tasks of the system, even before designing execution details [1, 7]. In the synthesis methods used so far, software and hardware parts are developed separately and then composed, which are resulting in cost increasing and decreasing quality and reliability of the final product.

Task scheduling is one of the most important issues occurring in the synthesis of operating systems responsible for controlling allocation of tasks and resources in computer systems.

The objective of the research is to present the concept of combined approach to the problem of system synthesis, i.e. a coherent solution to task scheduling and resource partition problems. In this paper was presented the model and the approach, which are

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_16,

proposals allowing synergic design of hardware and software for performing operations of the computer system [5].

Another important issue that occurs in designing computer systems is assuring their fault-free operation. Such synthesis concentrates on developing dependable and fault-tolerance architectures and constructing dedicated operating systems for them. In these systems an appropriate strategy of self-testing during regular exploitation must be provided. In general, fault tolerance architectures of computer systems are multiprocessor ones. The objective of operating systems in multiprocessor systems is scheduling tasks and their allocation to system resources. For fault tolerance operating system, this means scheduling usable and additionally testing tasks, which should detect errors of executive modules, in particular processors [9, 10].

Modeling fault tolerance systems consists of resource identification and task scheduling problems which are both NP-complete [2, 6]. Algorithms for solving such problems are usually based on heuristic approaches. The objective of this paper is to present the concept of combined approach to the problem of fault tolerant system synthesis, i.e. a coherent solution to task scheduling and resource assignment problems. The solution includes also the system testing strategies.

2 The Classical Process of Computer System Synthesis

The term synthesis of computer systems affects hardware and software [4, 7]. The classical synthesis process consists of the following general stages (Figure 1):

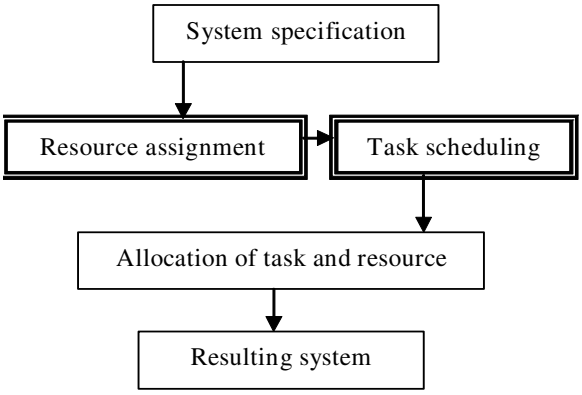


Fig. 1. The classical process synthesis of computer system

1. Specification of the system in terms of its functions and behaviour – requirements and constraints analysis. The system description in a high-level language, abstracting from a physical implementation.
2. Selection of the system architecture and control.

3. Resource partitioning – architecture development.
4. Task scheduling – system control development.

The system being constructed consists of hardware elements and software components performed by selected hardware modules. The system is specified by a set of requirements to be performed. The requirements specify also the expected system reliability.

In general, each requirement may be satisfied by hardware elements or software components executed by general processors and memories. Obviously, at this stage of design, one must take into account appropriate system constraints and criteria of dependable and optimal system operation. Accordingly, the key issue in the synthesis is efficient partitioning of system resources due to their hardware and software implementation, providing assertion of all requirements and the minimum implementation cost.

Such partitioning methodology may accept, as a starting point, assignment of the hardware implementation to all system functions and further optimization of project costs, search for possibilities of replacing certain tasks realized by hardware with their software equivalents. Other methods of the resources partitioning start with an exclusive software implementation and further search for implementation of certain tasks by hardware. In both approaches the objective is optimization of the implementation cost of the same tasks, i.e. in particular minimization of the execution time by specialized hardware. Obviously the requirements and constraints, especially those regarding time and dependability, have decisive influence upon selection of necessary hardware components. The measure for an efficient implementation of a multiprocessor system is the degree of its modules utilization, minimized idle-time of its elements and maximized parallel operation of its elements [12].

A non-optimum system contains redundant modules or modules that are excessively efficient in comparison to the needs defined by the tasks what, consequently, increases the system cost. In high-level synthesis, the optimization of the designed system costs and speed is usually an iterative process, requiring both changes in the architecture and task scheduling.

That is, why an optimum system may be created as a compromise between the system control algorithm and its hardware organization.

3 The Model for the Problem of Scheduling in Dependable Computer System Synthesis

System synthesis is a multi-criteria optimization problem. The starting point for constructing our approach to the issues of hardware and software synthesis is the deterministic theory of task scheduling [3]. The theory may serve as a methodological basis for fault tolerance multiprocessor systems synthesis. Accordingly, decomposition of the general task scheduling model is suggested, adequate to the problems of fault tolerance system synthesis.

From the control point of view such a model should take into account the tasks, which may be either preemptable or nonpreemptable. These characteristics are defined according to the scheduling theory. Tasks are preemptable when each task can be interrupted and restarted later without incurring additional costs. In such a case the schedules are called to be preemptive. Otherwise, tasks are nonpreemptable and schedules nonpreemptive. Preemptability of tasks in our approach cannot be a feature of the searched schedule – as in the task scheduling model so far. The schedule contains all assigned tasks with individual attributes: preemptive, nonpreemptive. From the point of view of the system synthesis, the implementation of certain tasks from the given set must be nonpreemptible, for the other may be preemptible (what, in turn, influences significantly selection of an appropriate scheduling algorithm). The above approach allows for inclusion into the discussed set of tasks also the system functions that are operating in real time. They must be realized by the hardware elements, often specialized ones. In such cases, the model is relevant to the synthesis of multiprocessors computer system. The set of functions not only models software units realized by a general hardware (processors and memories), but may represent all functions of the system, including those which must be realized by specialized components (e.g. hardware testing tasks). Moreover, we wish to specify the model of task scheduling in a way suitable for finding optimum control methods (in terms of certain criteria) as well as optimum assignment of tasks to universal and specialized hardware components.

Accordingly, we shall discuss the system:

$$\mathbf{S = \{ R, T, C \}} \tag{1}$$

where: R – is the set of resources (hardware and software), T – is the set of the tasks (operations), C – is the set of optimization criteria.

Resources. We assume that processor set $P = \{P_1, P_2,\ldots, P_m\}$ consists of *m* elements and additional resources $A = \{ A_1, A_2,\ldots, A_p\}$ consist of *p* elements.

Tasks. We consider a set of *n* tasks to be processed with a set of resources. The set of tasks is divided into 2 subsets: $T^1 = \{T_1^1, T_2^1,\ldots, T_{n1}^1\}$, $T^2 = \{T_1^2, T_2^2,\ldots, T_{n2}^2\}$, where $n = n1 + n2$ [1, 2]. Each task T_i^1 ($i = 1,2,\ldots, n1$) requires one arbitrary processor for its processing and its processing time is equal to t_i^1 . Similarly, each task T_i^2 ($i =1, 2,\ldots, n2$), requires 2 arbitrary processors simultaneously for its processing during a period of time whose length is equal to t_i^2. A schedule is called feasible if, besides the usual conditions, each task T_i^1 is processed by one processor and each task T_i^2 is processed by 2 processors at a time. A feasible schedule is optimal, if it is to minimized length. Each task is defined by a set of parameters:

- Resource requirements. The task may additionally require *j* units of resource A_i.
- Execution time.
- Ready time and deadline.
- Attribute - preemptable or nonpreemptable.

The tasks set may contain defined precedence constraints represented by a digraph with nodes representing tasks, and directed edges representing precedence constraints.

If there is at least one precedence constraint in a task set, we shall refer it to as a set of dependent tasks; otherwise they are a set of independent tasks.

Let us assume that dependable system resources include general parallel processors and specialized processors. As for tasks, we assume one-processor tasks used for modeling usable preemptable/nonpreemptable and dependent tasks, and two-processor tasks, for which we assume time and resource non-preemptiveness.
Two-processor tasks model the testing tasks (e.g. one processor checks the other). Testing tasks may be dependent on the defined time moments of readiness to perform and to complete assigned tasks. Two-processor tasks may realize a defined strategy of testing a dependable system.

Optimality criteria. As for the optimality criteria for the system being designed, we shall assume its minimum cost and maximum operating speed.

4 Coherent Process of Dependable System Synthesis

Modeling the joint search for the optimum task schedule and resource partition of the designed system into hardware and software parts is fully justified. Simultaneous consideration of these problems may be useful in implementing optimum solutions, e.g. cheaper hardware structures. With such approach, the optimum task distribution is possible on the universal and specialized hardware and defining resources with maximum efficiency. We propose the following schematic diagram of a coherent process of fault tolerance systems synthesis (Figure 2).

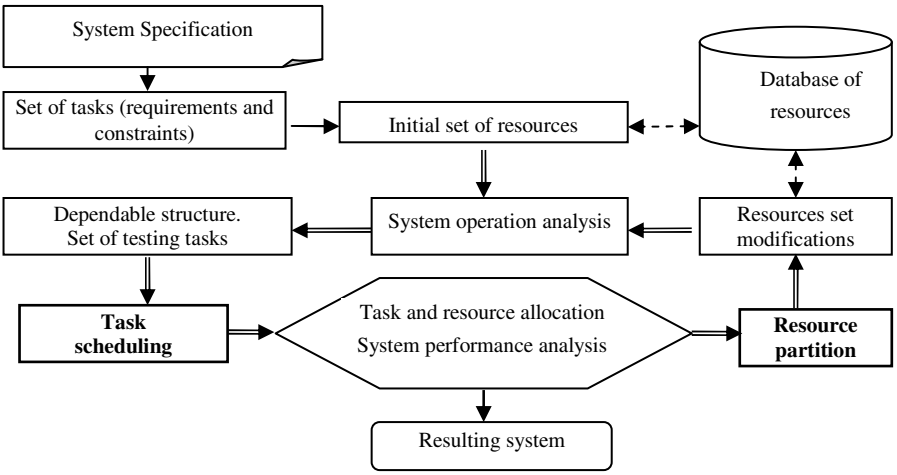


Fig. 2. The process coherent synthesis of dependable computer system

The suggested coherent analysis consists of the following steps:

1. Specification of requirements for the system,

2. Specification of tasks,
3. Assuming the initial values of resource set,
4. Defining testing tasks and the structure of dependable system,
5. Testing strategy selection,
6. Task scheduling,
7. Evaluating the operating speed and system cost, multi-criteria optimization,
8. The evaluation should be followed by a modification of the resource set, a new system partitioning into hardware and software parts and an update of test tasks and test structure (step 5).

In this approach a combined search for optimal resources partition and optimal tasks scheduling occur. Iterative calculations are executed till satisfactory design results are obtained – i.e. optimal system structure, dependable level and schedule.

5 Example of Dependable System Synthesis

We will consider the synthesis of the computer system which is testing electronic circuits type of computer motherboard. A system will be optimized as regards cost (cost minimization) and execution speed (schedule length minimization). Test system for such motherboards will be implemented in the same hardware structures: first in the non-dependable system and second in the dependable system.

5.1 System Specification

Testing system should realize tasks, which can be presented by a digraph (Fig. 3). Testing of the package should be realized in the following order: T_0 test of power supply circuits, T_1 test of processor, T_2 test of buses, T_3 test of interrupt controller, T_4 test of graphics controller, T_5 test of memory, T_6 test of disc controller, T_7 test of network controller, T_8 test of audio-video controller, T_9 test of program executing. Precedence constraints of the functions, which are to be realized by the system, ensue from package test procedures.

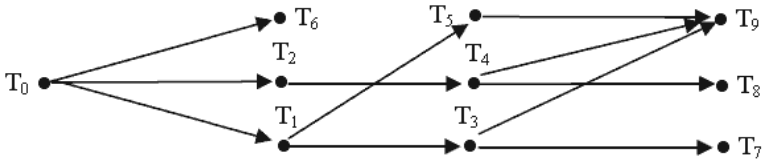


Fig. 3. Test executing sequence

Test of power supply circuits is the digraph root. Next, there are tests of processor and test of system buses. Interrupt controller can be tested after checking the processor. Test of network controller requires that interrupt controller is in working order and therefore T_7 task follows T_3 task. Next package test procedure realizes memory test (T_5) with processor (T_1) in working order (because previously checked), and tests of controllers of disc and graphics. Graphics controller checked and in working order

makes it possible to realize the following test of audio-video controller. Eventually, test of checking cooperation between processor and memory is being executed (T_9). We will assume executing times of tasks the following: $t_0 = 3$, $t_1 = 2$, $t_2 = 2$, $t_3 = 1$, $t_4 = 3$, $t_5 = 3$, $t_6 = 4$, $t_7 = 3$, $t_8 = 1$, $t_9 = 1$.

As an optimal criterion for projecting system we will assume minimization of executing time for all testing tasks – minimization of testing tasks schedule length. The second considered optimal criterion is system cost which takes into consideration cost of all hardware resources. Cost of system is:

$$C_S = m * C_P + i * C_M \quad (2)$$

where: C_P – processor cost, C_M – memory cost, i – number of memory modules equal number of tasks assigned to general processors, m – the number of identical parallel processors.

Additional requirements of designing system, we will assume as the following:

RI There is necessary a deadline of all tasks without delays executing which equals 13 time units.

RII There is necessary execute task T_6 (because disc controller usually operating in real-time) nonpreemptable.

RIII There is necessary a critical line for T_6 task which equals 9 time units.

RIV There is desirable a deadline, executing all processes without delay, equal 9 time units.

5.2 Project of Structure and Schedule in Non-dependable System

If we consider two identical and parallel processors, then this optimal schedule fulfills requirement RI that will realize *Muntz and Coffman* algorithm [3] (Fig. 4).

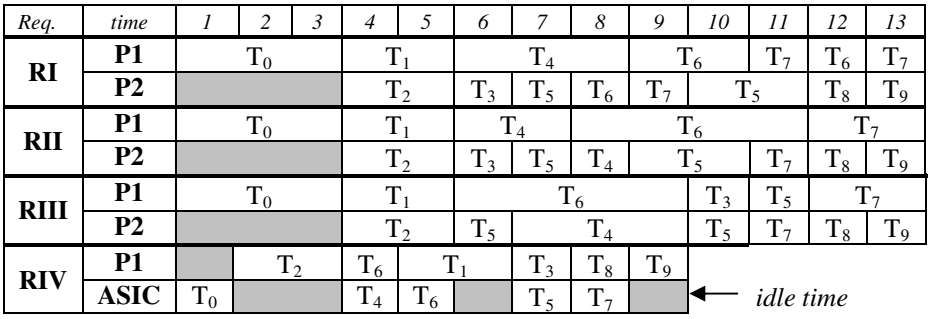


Fig. 4. Realized requirements RI, RII, RIII, RIV

Cost of this system – with assumption that every processor needs one memory unit, equals $C_S = 2*C_P + 10*C_M$. Taking into consideration requirement RII it is necessary to correct tasks schedule. System cost does not change. For requirement RIII realization there is necessary further correction of tasks schedule. System cost does not

change, too. At last it turns out; that all requirements (include RIV) cannot be realized on two processors. We will apply specialized resource, which can execute tasks: T_0, T_4, T_5, T_6, T_7 with a triple speed (as compared to the general processor) and its cost is C_{ASIC}. Resources structure and processor schedule we are showing also in Fig. 4. System cost equals $C_S = C_P + 6 * C_M + C_{ASIC}$

5.3 Project of Structure and Schedule in Dependable System

The cost of the system should be as low as possible, and the architecture conformant with the fault tolerance system model is required, with two-processor testing tasks. We shall assume the following labeling for processor testing tasks - T_{gh}, where P_g processor is testing (checking) P_h processor.

Implementing the system satisfying the requirement RI, the architecture of a fault tolerance system was shows in Figure 5 – Structure 1.

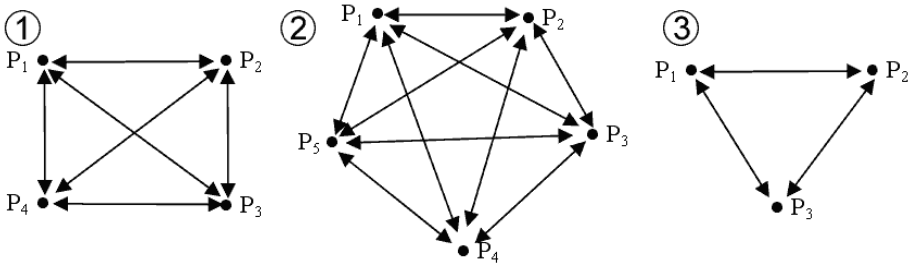


Fig. 5. Structure **1:** Processors and two-processor testing tasks in a dependable four-processor structure: T_{12}, T_{13}, T_{14}, T_{23}, T_{24}, T_{21}, T_{34}, T_{31}, T_{32}, T_{41}, T_{42}, T_{43}; **Structure 2**: Processors and two-processor testing tasks in a dependable five-processor structure: T_{12}, T_{13}, T_{14}, T_{15}, T_{21}, T_{23}, T_{24}, T_{25}, T_{31}, T_{32}, T_{34}, T_{35}, T_{41}, T_{42}, T_{43}, T_{45}, T_{51}, T_{52}, T_{53}, T_{54}; **Structure 3**: Processors and two-processor tasks in a dependable three-processor system with a specialized ASIC processor - T_{12}, T_{13}, T_{23}, T_{21}, T_{31}, T_{32}.

For such architecture, the optimum tasks schedule, guaranteeing the requirement RI, has been shown in (Fig. 6).

Taking into account the requirement RII, the following correction is done to the task schedule. Thus, we obtain the schedule shown in requirement RII. The system architecture and costs remain unchanged.

The next requirement RIII is reflected in a corrected schedule presented in requirement RIII.

Considering the RIV requirement, the system structure change is necessary. Two variants of the structure shall be proposed. The first structure consists of five identical parallel processors, with two-processor testing tasks Figure 5 (Structure 2). Task schedule in such structure is depicted in Figure 7. In the second variant, a specialized module (ASIC) was applied, that may perform the tasks: T_0, T_4, T_5, T_6 and T_7 with a triple speed (as compared with the general processor). The system structure and schedule are shown in Figure 5 (Structure 3) and Figure 7, respectively. The general

processor completes processing of usable tasks in 9 time units, while ASIC processor completes performing its function in 8 time units. Accordingly, the required deadline was reached in 9 units.

Req.	*time*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
RI	**P1**	T_{12}	T_0	T_0	T_{13}	T_1	T_{31}	T_{14}	T_{21}	T_7	T_{41}	T_7	T_6	T_{12}
	P2	T_{12}	T_{23}		T_1	T_{24}	T_4	T_4	T_{21}	T_{32}	T_6	T_{42}	T_8	T_{12}
	P3	T_0	T_{23}	T_{34}	T_{13}	T_2	T_{31}	T_5	T_4	T_{32}	T_5	T_5	T_{43}	T_7
	P4			T_{34}	T_2	T_{24}	T_3	T_{14}	T_6	T_6	T_{41}	T_{42}	T_{43}	T_9
RII	**P1**	T_{12}	T_0	T_0	T_{13}	T_1	T_{31}	T_{14}	T_{21}	T_7	T_{41}	T_6	T_7	T_{12}
	P2	T_{12}	T_{23}		T_1	T_{24}	T_4	T_4	T_{21}	T_{32}	T_6	T_{42}	T_8	T_{12}
	P3	T_0	T_{23}	T_{34}	T_{13}	T_2	T_{31}	T_5	T_4	T_{32}	T_5	T_5	T_{43}	T_7
	P4			T_{34}	T_2	T_{24}	T_3	T_{14}	T_6	T_6	T_{41}	T_{42}	T_{43}	T_9
RIII	**P1**	T_{12}	T_0	T_0	T_{13}	T_1	T_{31}	T_{14}	T_{21}	T_7	T_{41}	T_4	T_7	T_{12}
	P2	T_{12}	T_{23}		T_1	T_{24}	T_6	T_6	T_{21}	T_{32}	T_4	T_{42}	T_8	T_{12}
	P3	T_0	T_{23}	T_{34}	T_{13}	T_2	T_{31}	T_5	T_4	T_{32}	T_5	T_5	T_{43}	T_7
	P4			T_{34}	T_2	T_{24}	T_3	T_{14}	T_6	T_6	T_{41}	T_{42}	T_{43}	T_9

Fig. 6. Tasks schedule satisfying the requirement RI in a four-processor system (Req. RI), the requirements RI and RII in a four-processor system (Req. RII), the requirements RI, RII, and RIII in a four-processor system (Req. RIII) – Structure 1

For the requirements: RI, RI+RII, RI+RII+RIII : $m = 4$, $n_u = 10$, $p = 0$. For the requirements RI+RII+RIII+RIV, in the first variant, $m = 5$, $n_u = 10$, $p = 0$. In the second variant, where one ASIC processor is applied, $m = 3$, $n_u = 6$ and $p = 1$.

St.	*time*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
2	**P1**	T_{12}				T_{13}	T_3	T_4	T_{14}	T_9	T_{15}	T_{21}		
	P2	T_{12}	T_{23}		T_2	T_2	T_{24}	T_5	T_5	T_{25}	T_8	T_{21}	T_{32}	
	P3		T_{23}	T_{34}	T_6	T_{13}	T_5	T_{35}	T_4	T_4	T_9		T_{32}	T_{43}
	P4	T_0	T_0	T_{34}	T_{45}	T_1	T_{24}	T_6	T_{14}	T_7	T_7			T_{43}
	P5			T_0	T_{45}	T_6	T_6	T_{35}	T_7	T_{25}	T_{15}			
3	**P1**	T_{12}	T_2	T_{13}	T_{21}	T_1	T_{31}	T_{12}	T_8	T_{13}				
	P2	T_{12}	T_{23}	T_2	T_{21}	T_{32}	T_1	T_{12}	T_{23}	T_{13}				
	P3		T_{23}	T_{13}	T_6	T_{32}	T_{31}	T_3	T_{23}	T_9				
	ASIC	T_0			T_4	T_6		T_5	T_7					

Fig. 7. Tasks schedule satisfying the requirements RI, RII, RIII and RIV: in the five-processor system – Structure 2 and in the three-processor system with the specialized processor – Structure 3

The cost of the developed system shall be estimated as follows. If we assume that each usable task performed by a universal processor needs one memory unit dedicated to such task, and task assigned to ASIC processor do not need dedicated memory units the system cost is:

$$C_S = m * C_P + n_u * C_M + p * C_{ASIC} \quad (3)$$

where: m – the number of identical parallel processors, n_u – the number of tasks assigned to general processors, p – the number of specialized ASIC processors devoted for processing remaining $(n - n_u)$ tasks.

6 Conclusions

In this paper an attempt of combined approach to high-level system synthesis is presented. This synthesis is directed to systems with self-testing of their main resources, i.e. the processors.

Such approach in designing dependable systems is applied self testing multiprocessor tasks. The following system optimization criteria are accepted: minimum operating time and minimum cost. Combined implementation of resource partitioning and task scheduling can provide better solutions that those obtained with separate resource partitioning and task scheduling. The synergic solution is a result of cooperation between the scheduling algorithms and the algorithm responsible for resource partition. This conforms the correctness of the concept of joint development of hardware and software parts in system design.

The model presented for coherent synthesis and the approach to the self testing allow a further research in this area. One may specify additional optimality criteria, e.g. minimum power consumption of the designed system (which is particularly significant for built-in and mobile systems) [11]. For the proposed system's relevance to real systems, one should take into account the processes of communication between resources and tasks, preventing resource conflicts, as well as extend the available resources sets, for example by programmable and configurable structures [10]. From a dependable system point of view, the testing and diagnosis scheme is significant for the system's fault tolerance. This work presents only one of the methods of providing self-testability of the designed system. Reliability is particularly significant for real time systems. That is why the synthesis should include the criterion of the task scheduling optimality before deadlines. The problem of the combined synthesis is a multicriteria optimization problem. Taking into account several criteria and the fact that optimization of one criterion results often in worsening of the second one, the Pareto optimization [12] can be a right solution providing a trade-off between multiple criteria. The optimum solution in such case shall be the one not dominated by any other solution in the whole solution space. There is no single optimum in such case, but a set of solutions trying to satisfy contradictory criteria. Thus, as result of a coherent system synthesis, we obtain a set of solutions that are optimal in the Pareto sense.

The optimum solution in such case shall be the whole expanse of solutions. The above issues are now studied.

References

1. Aggoune, R.: Minimizing the makespan for the flow shop scheduling problem with availability constraints. Eur. J. Oper., Res. 153, 534–543 (2004)
2. Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G., Węglarz, J.: Handbook on Scheduling, From Theory to Applications. Springer, Heidelberg (2007)

3. Błażewicz, J., Ecker, K., Plateau, B., Trystram, D.: Handbook on Parallel and Distributed Processing. Springer, Heidelberg (2000)
4. Dick, R.P., Jha, N.K.: COWLS: Hardware-Software Co-Synthesis of Distributed Wireless Low-Power Client-Server Systems. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 23(1), 2–16 (2004)
5. Drabowski, M., Wantuch, E.: Coherent concurrent task scheduling and resource assignment in dependable system design, Advances in Safety and Reliability – ESREL 2005. In: Proceedings of the European Safety and Reliability Conference. Taylor & Francis (2005)
6. Golub, M., Kasapovic, S.: Scheduling multiprocessor with genetic algorithms. In: Proceedings of the IASTED Applied Informatics Conference, Innsbruck (2002)
7. Oh, H., Ha, S.: Hardware-software synthesis of multi-mode multi-task embedded systems with real-time constraints. In: Proceedings of the IEEE/ACM Conference on Hardware Software Codesign, Estes Park, Colorado, pp. 133–138 (2002)
8. Kordon, F., Luqi, A.: An Introduction to Rapid System Prototyping. IEEE Trans. on Software Engineering 28(8), 817–821 (2002)
9. Yhang, Z., Dick, R., Chakrabarty, A.: Energy-aware deterministic fault tolerance in distributed real-time embedded systems. In: 41st Proc. Design Automation Conf., Anaheim, California, pp. 550–555 (2004)
10. Yhang, Z., Dick, R., Chakrabarty, A.: Energy-aware deterministic fault tolerance in distributed real-time embedded systems. In: 41st Proc. Design Automation Conf., Anaheim, California, pp. 550–555 (2004)
11. Ziegenbein, D., Richter, K., Ernst, R., Thiele, L., Teich, J.: SPI – A System Model for Heterogeneously Specified Embedded Systems. IEEE Trans. on VLSI Systems 10(4), 379–389 (2002)
12. Ziegenbein, D., Jersak, M., Richter, K., Ernst, R.: Breaking Down Complexity for Reliable System-Level Timing Validation. In: Proc. of the Electronic Design Process Workshop (2002)

Using Simulation to Evaluate Dynamic Systems with Weibull or Lognormal Distributions

Ernest Edifor, Neil Gordon, Martin Walker, and Yiannis Papadopoulos

Department of Computer Science, University of Hull, Hull, UK
{e.e.edifor@2007.,n.a.gordon@,martin.walker@,
y.i.papadopoulos@}hull.ac.uk

Abstract. Most techniques for quantitatively analysing the temporal fault trees of safety-critical systems are used with the assumption that the systems under study have exponentially distributed component failures. However, it is not impossible for real world systems to have various components with different failure distributions. This paper presents a simulation approach – Monte Carlo – for modelling, simulating and estimating the total system failure of state-of-the-art dynamic systems featuring Weibull or lognormal distributions. The proposed techniques have been formulated using the time-to-failure of these distributions to model temporal behaviours; they can be extended to model systems with other distributions.

Keywords: Safety-Critical systems, Temporal Fault Trees, Exponential Distribution, Weibull Distribution, Lognormal Distribution, Monte Carlo Simulation.

1 Introduction

The advent and acceptance of technological systems in this age calls for high levels of reliability of these systems. This is because the failure of some of these systems can have catastrophic effects on both their environment and human life. Due to the effects such systems can have if they fail, their reliability is a core requirement. This is a domain covered by reliability engineering.

Fault Tree Analysis (FTA) [1] is a deductive technique used in reliability engineering for investigating how combinations of components failures can propagate to cause an entire system to fail. FTA is primarily analysed either quantitatively (probabilistically) or qualitatively (logically). Qualitative analysis involves the determination of combinations of basic faults that can lead to a total failure of a system (known as the top-event) under consideration. The smallest combinations of basic component faults which lead to the occurrence of the top-event are known as Minimal Cut Sets (MCS). Quantitative analysis, on the other hand, provides numeric quantities representing the probability that the top-event will occur within a specific time given the basic component failure data or the relative importance between MCSs or basic components and their contribution to the top-event occurrence.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_17,

FTA has a major limitation: an inability to capture the sequential order in which basic events occur. This dynamic behaviour inherent in some safety-critical systems cannot be overlooked when the systems are being designed and analysed. Doing so can result in the inaccurate estimation of MCS and top-event probability [2]. Various efforts [3, 4] have been made to solve this problem. Among these is the newly developed Pandora [5-7] based on temporal fault trees.

Pandora evaluates temporal fault trees with the use of the Priority-AND (PAND), Priority-OR (POR), Simultaneous-AND (SAND) and parameterized-SAND (pSAND) gates. It provides both logical [5-7] and probabilistic [8-9] evaluations of temporal fault trees. The probabilistic evaluations of temporal fault trees using Pandora is restricted to dynamic systems with exponentially distributed component failures. However, it is well known that not all systems demonstrate this failure distribution [10].

The purpose of this paper is to provide a simulation approach for evaluating dynamic safety-critical systems with either Weibull or lognormal distributions using Pandora. A Weibull distribution is described as the most popular component life distribution used in engineering [10] and can be used for modelling acceptance sampling, warranty analysis, wear or corrosion modelling, whilst the lognormal distribution can be used in modelling many component life distributions and component repair time distributions [10]. The propositions in this paper are holistic and more 'system-friendly' for evaluating various types of dynamic high-consequence systems.

It is assumed in this paper that components are independent and their failures are non-repairable. The remaining sections of this paper are structured as follows: chapter two provides a background review of temporal fault tree analysis and failure distributions. Chapter three provides Monte Carlo simulations for modelling systems with Weibull or lognormal distributions. In chapter four, we apply the techniques proposed on an aircraft fuelling system and discuss the results. Finally, conclusions are drawn in chapter five.

2 Background

2.1 Temporal Fault Tree-Pandora

Pandora [5-7] is an extension of FTA [1] which uses three temporal gates – PAND, POR, and SAND – and over eighty novel temporal laws in addition to the original Boolean gates – AND and OR – and laws to provide a comprehensive logical analysis of temporal fault trees. PAND, represented by '<', stands for "Priority-AND" and it is true if and only if its input events occurs strictly one before another. SAND, represented by '&', stands for "Simultaneous-AND" and is true only when all its input events occur at exactly the same time. POR, represented by 'l' means "Priority-OR" and it occurs if its first input event occurs before its second input event, but the second event is not required to occur. The original meaning and semantics of Boolean AND and OR are maintained; '.' for AND and '+' for OR.

More recently, a new temporal gate, parameterized-SAND (pSAND) [9], has been included in the quantitative analysis of Pandora. The pSAND gate is used to represent

nearly simultaneous situations – where an output event is triggered when its input events occur only within a relatively small duration of interval.

The qualitative analysis of temporal fault trees using Pandora can be undertaken with one (or both) of two techniques – Euripides or Archimedes [7]. The qualitative analysis produces a set of smallest combinations of events that can lead to the top-event; these are called Minimal Cut Sequences (MCSQs). The description of the qualitative analysis of Pandora is beyond the scope of this paper. Quantitative analysis of temporal fault trees of systems with exponential distribution using both analytical and simulation approaches have been provided in previous literature [8-9]. For the sake of revision, we state the analytical techniques for all the gates used in Pandora; $P(X)$ is the probability of event X occurring between zero and a time, t, and n is the number of events.

$$P(X_1 + X_2 + \cdots + X_{n-1} + X_n)\{t\} = 1 - \prod_{j=1}^{n}\left(1 - \prod_{i=1}^{m_j} P\left(MCSQ_{i,j}\right)\{t\}\right) \tag{1}$$

where, m_j is a MCSQ with m combinations of basic events

$$P(X_1 \cdot X_2 \cdot \ldots \cdot X_{n-1} \cdot X_n)\{t\} = \prod_{k=1}^{n} P(X_k)\{t\} \tag{2}$$

$$P(X_1 < X_2 < \cdots < X_{n-1} < X_n)\{t\} = \prod_{i=1}^{n} \lambda_i \sum_{k=0}^{n}\left[\frac{e^{(a_k t)}}{\prod_{\substack{j=0 \\ j\neq k}}^{n}(a_k - a_j)}\right] \tag{3}$$

Where $a_0=0$ and $a_m= -\sum_{j=1}^{m}\lambda_i$ for $m>0$.

$$P(X_1|X_2|\ldots|X_{n-1}|X_n)\{t\} = \frac{\lambda_1\left(1 - \left(e^{-(\sum_{i=1}^{n}\lambda_i)t}\right)\right)}{\sum_{i=1}^{n}\lambda_i} \tag{4}$$

$$P(X_1 \&_d X_2 \&_d \ldots \&_d X_{n-1} \&_d X_n)\{t_0, t_1\} = \sum_{i=1}^{n}\left(F(X_i)\{t_0\}.\left(\prod_{\substack{j=1 \\ j\neq i}}^{n} F(X_i)\{t_0, t_1\}\right)\right) \tag{5}$$

Eqn. (1) and Eqn. (2) [1] are for evaluating the Boolean OR and AND respectively. Eqn. (3) [11] can be used in evaluating the PAND gate whilst Eqn. (4) [8] and Eqn. (5) [9] can be used for evaluating the POR and pSAND gates respectively. The SAND gate evaluates to zero and is ignored in this paper. The Monte Carlo simulations for estimating the pSAND, PAND and POR gates are presented in [9], [12] and [8] respectively. The increasing precedence order for quantitatively evaluating Pandora is:

pSAND > PAND > POR > AND > OR

2.2 System Failure Distributions

The Weibull distribution is used for modelling acceptance sampling, warranty analysis, wear modelling, and corrosion modelling maintenance and renewal and material strength [10]. Generally, it has two parameters – a scale parameter α and a shape parameter β. For any system S with a lifetime t the cumulative distribution function (CDF) can be represented by (6) [10].

$$F(S)\{t\} = 1 - e^{-\left(\frac{t}{\alpha}\right)^{\beta}} \tag{6}$$

The lognormal distribution is suitable for modelling many life distributions and components with repair time distributions and is appropriate for parameter variability and modelling elements in breakage processes [10]. For any system S with a life time t, mean of normal distribution μ, standard deviation of normal distribution σ and standard normal CDF Φ then the CDF can be given as (7) [10].

$$F(S)\{t\} = \Phi\left(\frac{\ln(t) - \mu}{\sigma}\right) \tag{7}$$

3 Simulating Systems with Different Distributions

As already mentioned, simulation for the exponential distributions have already been provided for the PAND, POR and pSAND gates. We present the modelling of all these gates for both the Weibull and lognormal distributions. Modelling the AND and OR gates are simple and therefore ignored. It must be noted that throughout this section, *F(a, b ,t)* and *TTF(a, b, r, t)*, are the CDF and TTF (time-to-failure) respectively for an event with α=a and β=b (for a Weibull distribution) or μ=a and σ=b (for a lognormal distribution); *r_X* is a random number representing the failure probability of event *X*. Before constructing the simulation conditions necessary for modelling temporal gates, we have to identify the time-to-failure (TTF) of Weibull and lognormal distributions. Given the CDF for a Weibull distribution in (6) and lognormal distribution in (7) the TTFs for both distributions will be (8) and (9) respectively.

$$TTF_{Weibull} = \alpha.\sqrt[\beta]{\log\left(\frac{1}{1-r}\right)} \tag{8}$$

$$TTF_{lognormal} = e^{\sigma.\Phi^{-1}(r)+\mu} \tag{9}$$

The Monte Carlo condition required for the PAND gate for two events, *x* and *y*, referred to as SimPAND, is:

```
r_x <= F(x_a, x_b, t) AND r_y <= F(y_a, y_b, t) AND
   TTF(x_a, x_b, r_x, t) < TTF(y_a, y_b, r_y, t)
```

In SimPAND, *x* and *y* would have to occur and *x* would have to occur before *y*.

The Monte Carlo conditions required for the POR gate for two events, *x* and *y* is referred to as SimPOR and given as:

```
r_x <= F(x_a, x_b, t)) AND r_y <= F(y_a, y_b, t) AND
   TTF(x_a, x_b, r_x, t) < TTF(y_a, y_b, r_y, t) OR
   (r_x <= F(x_a, x_b, t) AND r_y > F(y_a, y_b, t)
```

Meaning, if *x* and *y* occurs, *x* would have to occur before *y* or if *x* occurs, *y* would not occur.

The Monte Carlo conditions required for the pSAND gate for two events, *x* and *y* with a duration of interval, *d*, is referred to as SimSAND and is given by:

```
r_x <= F(x_a, x_b, t) AND r_y > F(y_a, y_b, t) AND
   r_y <=F(y_a, y_b, t+d)) OR (r_y <= F(y_a, y_b, t)AND
   r_x > F(x_a, x_b, t) AND r_x <= F(x_a, x_b, t+d)
```

SimSAND means *x* and *y* would have to occur and they would have to do so within *d*. The SimSAND, SimPAND and SimPOR conditions stated above can be used for evaluating both Weibull and lognormal distributions. The most difficult aspect of simulating temporal fault trees is when a MCSQ contains more than one temporal gate – for example $A<B|C$ which is equivalent to $(A<B)|C$. Clearly, one can use SimPAND to evaluate $A<B$. However, SimPAND returns a TRUE or FALSE value which cannot be used in modelling $(A<B)|C$ because the '|' evaluation will require a numeric value (not Boolean value) for the TTF evaluation in SimPOR. To solve this problem, the condition required for modelling $A<B|C$ can be modelled rather as:

```
(SimPAND(r_A, A_a, A_b, r_B, B_a, B_b, t) AND
   r_C <= F(C_a, C_b, t) AND TTF(B_a, B_b, r_B, t) <
   TTF(C_a, C_b, r_C, t)) OR (SimPAND(r_A, A_a, A_b,
   r_B, B_a, B_b, t) AND r_C > F(C_a, C_b, t))
```

Meaning, instead of evaluating a TTF comparison between '$A<B$' and 'C' during the POR evaluation, the comparison is done between *B* and *C* only. This is because, if $A<B$ has occurred, then *B* has occurred after *A* so only the TTF comparison between the time of occurrences of *B* and *C* can be used to evaluate '$A<B|C$'.

4 Case Study

To demonstrate the techniques presented in this paper, we consider an Aircraft Fuelling System (AFS) (Fig. 1). AFS operates with the help of 5 fuel storage tanks (positioned along the horizontal axes of the aircraft to maintain a balance across its entire body), 7 bi-directional pumps (embedded with speed sensors to feed the port and starboard engines), 13 valves (to allow fuel to flow in directions requested by the pumps), 6 flow meters (for measuring the volume of fuel flowing across them), 2 jettison points (for releasing fuel into the atmosphere during an in-flight emergency) and fuel pipes (that connect tanks to engines, refuelling point and jettison points and are interconnected with pumps, valves and flow meters).

AFS also has a Fuel Supply Control Unit (FSCU) that is a centralised computer for controlling the entire AFS by providing the following functions:

- Ensure the even distribution of fuel to all five tanks during refuelling mode by controlling the valves and directions of pumps and determining the amount of fuel in each tank using the level sensors.
- Ensure both port and starboard engines are evenly fed from appropriate tanks during consumption mode by controlling pumps, valves and flow metres to deliver the required amount of fuel to both engines to satisfy the demand thrust.
- Communicates the status of the AFS with other aircraft computing systems. This also involves receives commands from the cockpit or other computing systems to control AFS's components and executes the command accordingly.

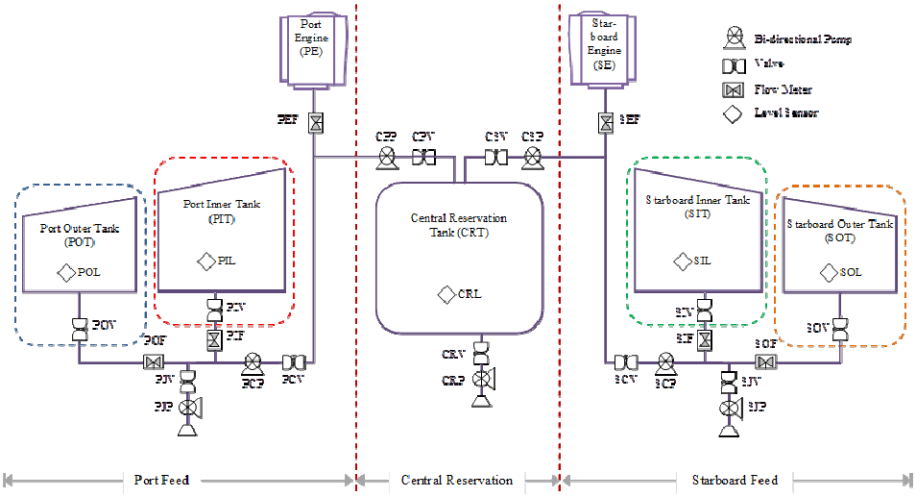


Fig. 1. Fuel distribution system

AFS is divided into three sub-systems: Port Feed (PF) sub-system, Central Reservation (CR) sub-system and the Starboard Feed (SF) sub-system. PF and SF are further divided into various subsystems. PF has the Port Outer Sub-system (POS) and the Port Inner Sub-system (PIS) whiles SF has the Starboard Outer Sub-system (SOS) and Starboard Inner Sub-system (SIS). The operations of AFS can be categorised in two main modes – refuelling and consumption – over the entire flight phase of the aircraft. The refuelling mode takes place before a flight (pre-flight) whiles the consumption mode occurs during a flight (in-flight), which includes the taxiing, take-off, climbing, cruising, approaching and landing phases. In this paper we only consider the consumption mode to demonstrate the proposed approach.

In normal operation, during consumption mode, PF tanks supply fuel to the port engine whiles the SF tanks feed the starboard engine. During this phase, the jettison valves are shut. The outer tanks are the primary sources of fuel to the engines while

the inner tanks are the secondary sources. Therefore to run SE, a demand is placed on the FSCU which ensures that SE's 'thirst' is satisfied by the SOT first. In this case, SIV is closed and the SIS is dormant. If SOS reads 'empty' or SOF fails low (meaning, due to a fault, it reads a lower value than expected), SOV is shut and control is directed to draw fuel from SIT instead; in this case SOV is closed and SIF is activated. If SIS reads 'empty' or SIF fails low, SIV and SCV are closed to deactivate the SF sub-system and SE is fed from the CRT, which serves as a tertiary backup. The same order of operations applies to the PF sub-system. To maintain a steady balance of the aircraft during the failure of any or all subsystems, FSCU communicates with another computing system, Aircraft Stability Control Unit (ASCU), which, with the use of other conduit/valve systems, ensures that fuel is evenly distributed across the horizontal axes of the plane. If, for any reason, ASCU is not able to maintain a steady balance across the aircraft, it activates appropriate jettison valves and pumps to release fuel into the atmosphere to achieve a steady balance. A detailed description and functionality of ASCU is outside the scope of this paper.

4.1 Qualitative Analysis of AFS

The failure of SF, CR and PF lead to the failure of the entire fuelling system. If either SF or PF fails in addition to CR, the aircraft will be in a degraded system – only one engine will be operational. If anyone of the sub-systems – POS, PIS, SOS or SIS – fails, the aircraft remains in an operational state: both engines function normally. If any of the feed systems – PF or SF – fails, the aircraft remains operational because CR will substitute the failed feed system. In this paper, we consider failure of the starboard feed (SF) system as the top-event. To model the failure data of AFS, the following abbreviation scheme is adopted: I-X means internal failure of a component X; O-X means omission of functionality of a component X; Hi-X means component X reads an erroneous high value.

Failure of the starboard engines is due to omission of functionality from Starboard Engine Flow meter (SEF). However,

O-SEF = Hi-SEF<O-SCV + O-SOS & O-SIS + O-SIS < O-SOS + O-SCV . O-CSP
O-SCV = I-SCV + (O-SOS . Hi-SIF) | O-SIS + O-SCP
O-SCP = I-SCP + O-SIS . O-SOS + Hi-SOF < O-SOS
O-CSP = I-CRL + I-CSV + I-CSP
O-SOS = I-SOV + I-SOL
O-SIS = I-SIV + I-SIL

Using temporal fault tree analysis – Pandora – the MCSQs are:

```
I-CSP.I-SCV + I-CRL.I-SCV + I-CSV.I-SCV + I-SCP.I-CSP +
I-CSP.I-SOL.I-SIL + I-CRL.I-SCP + I-SCP.I-CSV + I-
SIL&I-SOV + I-SOV&I-SIV + I-SOL&I-SIV + I-CRL.I-SOL.I-
SIL + I-SOL.I-SIL.I-CSV + I-SIV<I-SOV&I-SOL + I-SIV<I-
SOV|I-SOL + I-SIV<I-SOL|I-SOV + I-CRL.Hi-SIF.I-SOV +
Hi-SIF.I-SOV.I-CSV + Hi-SIF.I-CSP.I-SOL + Hi-SIF.I-
SOL.I-CSV + Hi-SOF<I-SOL.I-CSV + Hi-SOF<I-SOV.I-CSV +
```

```
Hi-SOF<I-SOL.I-CRL + Hi-SOF<I-SOV.I-CRL + Hi-SOF<I-
SOL.I-CSP + Hi-SOF<I-SOV.I-CSP + Hi-SIF.I-SOV.I-CSP
```

4.2 Quantitative Analysis of AFS

Before any quantitative analysis of AFS is evaluated, basic component failure data are assigned to its components in Table 1. These component failure data for various distributions are not necessarily related; λ is the constant failure rate per hour for exponential distribution, α and β are the scale and shape parameters for Weibull distribution and μ and σ are the mean and standard deviation for a lognormal distribution. Table 2 contains various *d* values for all pSAND gates. Tables 3, 4 and 5 are the top-event probabilities with various lifetimes for the exponential, Weibull and lognormal distributions respectively. 'S-MC' represents results for standard Monte Carlo simulation – no importance sampling – with $1x10^6$ trials whilst 'Exact' represents the result from analytical techniques – where there exist; 'NA' otherwise.

Table 1. Component Failure Data

Component	λ	α	β	μ	σ
I-SCP	5.84267E-5	3760	3.8	5.0235	1.1652
I-CSP	5.84267E-5	3760	3.8	5.0235	1.1652
I-SOV	1.65633E-3	535	0.7	7.0245	3.5152
I-SIV	1.65633E-3	535	0.7	7.0245	3.5152
I-CSV	1.65633E-3	535	0.7	7.0245	3.5152
I-SCV	1.65633E-3	535	0.7	7.0245	3.5152
I-CRL	2.21127E-6	4200	4.5	3.0125	1.1842
I-HiSOF	4.06861E-5	3510	3.8	6.0015	2.9332
I-HiSIF	4.06861E-5	3510	3.8	6.0015	2.9332
I-SOL	3.31774E-5	3490	3.2	8.0548	1.5122

Table 2. pSAND Time Intervals

CSQ	Intervals (d)/second
I-SIL&$_d$I-SOV	0.1
I-SOV&$_d$I-SIV	0.2
I-SOL&$_d$I-SIV	0.3
I-SOV&$_d$I-SOL	0.4

Table 3. Top-Event Probabilities using Exponential Distribution

System Lifetimes (*t*)	Exact	S-MC	% Error
1hr	4.3396207E-06	4.0000000E-06	7.8260E+00
100hr	3.6771307E-02	3.6429000E-02	9.3091E-01
10000hr	9.9999999E-01	1.0000000E+00	7.4732E-07
100000hr	1.0000000E+00	1.0000000E+00	0.0000E+00

From Table 3, the results of the analytical approach and Monte Carlo simulation estimation are similar. However, Tables 4 and 5 have no results for the analytical solution because they have not yet been developed.

Table 4. Top-Event Probabilities using Weibull Distribution

System Lifetimes (t)	Exact	S-MC	% Error
1hr	NA	2.1100000E-04	NA
100hr	NA	1.0338100E-01	NA
10000hr	NA	1.0000000E+00	NA
100000hr	NA	1.0000000E+00	NA

Table 5. Top-Event Probabilities using Lognormal Distribution

System Lifetimes (t)	Exact	S-MC	% Error
1hr	NA	9.2400000E-04	NA
100hr	NA	5.5378900E-01	NA
10000hr	NA	9.9999600E-01	NA
100000hr	NA	1.0000000E+00	NA

Tables 4 and 5 are the top-event probabilities for various lifetimes using the Weibull and lognormal distributions respectively. From both results it is obvious that with increasing time, the probability of the top-event occurring increases; the top-event for all distributions is likely to occur after 10000 hours of AFS operation.

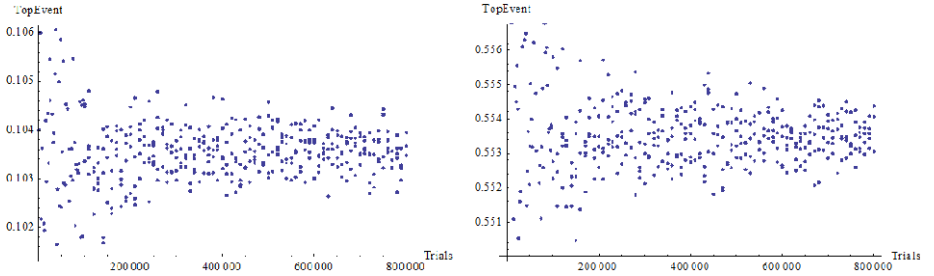


Fig. 2. S-MC Estimates for Weibull (left) and lognormal (right) Distribution at t=100hrs

The left and right images in Figure 2 are plots for checking the convergence of Weibull and lognormal distributions respectively when t=100 hrs. From Fig. 2(left) it can be seen that the results converge towards 0.1034 as seen in Table 4 and in Fig. 2 (right) the estimates converge towards 0.5538 as seen in Table 5. The component failure data used in all experiments are relatively small mimicking a more realistic system so a large number of trials have been used to achieve convergence. Future improvements may be possible using importance/dynamic sampling techniques.

5 Conclusion

Temporal fault trees for safety-critical systems can be analysed using both deterministic/analytical and simulation approaches. Many analytical approaches are done with the consideration that system component failures are exponentially distributed. However, many dynamic state-of-the-art systems exhibit other distributions such as the Weibull or lognormal distributions. We have presented Monte Carlo simulations for modelling systems with Weibull or lognormal distributions and applied the technique to an aircraft fuelling system. This paper has two very important benefits. Firstly, it has proven that simulation can be used for modelling the temporal fault trees of high-consequence systems where analytical approaches have not yet been developed or are impracticable. Secondly, it makes possible the modelling, simulation and probabilistic evaluation of dynamic systems with Weibull or lognormal distributions.

Implementing a dynamic stopping technique or importance sampling in the Monte Carlo simulation in this paper will hopefully reduce the computational resources needed to run the simulations. A possible extension of this paper will be the application of the techniques in other areas of study other than reliability engineering.

Acknowledgement. This work was partly supported by European project MAENAD (FP7 grant agreement number 260057).

References

1. Vesely, W.E., Stamatelatos, M., Dugan, J.B., et al: Fault Tree Handbook with Aerospace Applications, NASA Office of Safety and Mission Assurance, Washington DC (2002)
2. Merle, G., Roussel, J.: Algebraic Modelling of Fault Trees with Priority AND Gates. In: IFAC Workshop on Dependable Control of Discrete Systems, pp. 175–180 (2007)
3. Dugan, J., Bavuso, B., Boyd, S.J., Dynamic Fault-Tree, M.A.: for Fault-Tolerant Computer Systems. IEEE Transactions on Reliability 41(3), 363–376 (1992)
4. Merle, G.: Algebraic Modelling of Dynamic Fault Trees, Contribution to Qualitative and Quantitative Analysis: Dissertation, Décole Normale Supérieure De Cachan (2010)
5. Tang, Z., Dugan, J.: B.: Minimal Cut Set/Sequence Generation for Dynamic Fault Trees. In: Reliability And Maintainability Symposium (RAMS), Los Angeles, pp. 26–29 (2004)
6. Walker, M., Papadopoulos, Y.: Synthesis and Analysis of Temporal Fault Trees with PANDORA: The Time of Priority AND Gates. In: Nonlinear Analysis Hybrid Systems, vol. 2, pp. 368–382 (2008)
7. Walker, M.: Pandora: A Logic for the Qualitative Analysis of Temporal Fault Trees: Dissertation, University of Hull (2009)
8. Edifor, E., Walker, M., Gordon, N.: Quantification of Priority-OR Gates in Temporal Fault Trees. In: Ortmeier, F., Lipaczewski, M. (eds.) SAFECOMP 2012. LNCS, vol. 7612, pp. 99–110. Springer, Heidelberg (2012)

9. Edifor, E., Walker, M., Gordon, N.: Quantification of Simultaneous-AND Gates in Temporal Fault Trees. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 141–151. Springer, Heidelberg (2013)
10. O'Connor, A.N.: Probability Distributions Used in Reliability Engineering, Reliability Information Analysis Center, pp. 40–80 (2011)
11. Fussel, J., Aber, B., Rahl, E.F., On, R.G.: the Quantitative Analysis of Priority-AND Failure Logic. IEEE Transactions on Reliability R-25(5), 324–326 (1976)
12. Rao, D.K., et al.: Dynamic Fault Tree Analysis Using Monte Carlo Simulation in Probabilistic Safety Assessment. RESS 94(4), 872–883 (2008)

FSM Simulation of Cryptographic Protocols Using Algebraic Processor

Alexander Frolov and Alexander Vinnikov

National Research University Moscow Power Engineering Institute, 111250. Moscow, Krasnokazarmennaya, 14, Russian Federation
abfrolov@gmail.com, al.vin@bk.ru

Abstract. We study FSM model of cryptographic protocols that reflects both the system functionality and strategy of attacks and explored the fact that all data are divided into two classes: public transactions available to all parties and private data available to only party that inputted or originated them. In terms of this model the protocols FSM composition property and operation of composition of protocols FSM models are determined. This approach is supported by created software called algebraic processor that allows computer experiments to identify and demonstrate the leaks. We describe the structure and functionality of algebraic processor and some examples of attacked cryptographic protocols simulations.

Keywords: cryptographic protocol, finite state automaton (FSM), FSM model, algebraic processor, simulation of attacks, python, algebraic library, transmission intercepting, transmission delaying, transmission replacing, man-in-the-middle attack, parallel-session attack.

1 Introduction

Data protection in communicational networks is organized implementing cryptographic protocols. Cryptographic protocol is the collection of interacted cryptographic algorithms that ensure the data confidentiality, data integrity, data origin authentication, non-repudiation and other safety services. To justify these properties of protocols, various modeling tools are required. There are many approaches to cryptographic protocols modeling. There are known automatic methods of protocol security analyzing. This analysis allowed discovering security vulnerabilities of many cryptographic protocols [1,2]. The most advanced methods of protocol modeling are preserving their properties under superposition [2,3,4]. Such methods should take into account the specific features of cryptographic protocols, particularly the fact that in multiple executions of some protocol algorithms some fragments of algorithms are executed once [4], some data transfers from one party to another are made across the public and moreover could be broadcasted, while others are protected.

In this chapter, we propose the method of cryptographic protocols modeling which allows combined representation and analysis of protocol with intruder. United

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_18,

protocol is obtained by adding the intruder's algorithm to the protocol or by replacing some of protocol's algorithms with algorithms, executed depending on the fact of their corruption in standard or corrupt versions. The proposed method explores as a protocol model Moore finite states machine (FSM) that state set is the set of binary tuples characterizing the sets of broadcasted transactions during the protocol execution. Transition functions are defined by protocols and intruder's algorithms.

One should remark that currently FSM are used extensively for modeling secure cryptosystems and analysis reliability of computer programs [5,6,7].

Unlike [2,3,4,6] modeling purposes on the basis of the proposed model is not so much proof of security protocol as the identification and demonstration of possible leaks using a computer-model combining the protocol software and attacker program.

In section two we introduce the Moore automaton model of cryptographic protocol and consider some properties of those models.

In section three we consider the operation of protocol composition and FSM models of protocols with intruder and with corrupted parties.

In section four the structure and functionality of algebraic processor as software allowing experimental study of cryptographic protocols automaton models is shortly represented.

In conclusion the future tasks are discussed.

2 FSM Models of Cryptographic Protocols

In proposed model a cryptographic protocol is represented as a collection of programs implementing algebraic algorithms of the protocol. Each program is assigned to a particular network party and executed on a specific computer. Each party can possess one or more programs corresponding to protocol runs initialized by party. The input data of the protocol are indicating the name of the program to which they relate and may be public and admitting publication (for example, the description of the algebraic structure used in the protocol) or they may be confidential (for example, secret messages or keys). Confidential data also can be generated by the program when executed. It can be long-term or single-use security data. The input data for each program are input data of the protocol related to given program, as well as "shared" output data of all or some programs (data prefixed with "share."). Shared data are available for protocol programs of all parties. They are called transactions, as they serve the purpose of sharing between parties. In contrast, the confidential data are available only for programs of this party. Output of the protocol also are indicating the names of the programs that calculate them, and also considered to be confidential. Thus output data of each program may be transactions or part of protocol output data. Each program of the protocol can be called repeatedly. The long-term security data generated at a certain call in the subsequent executions of this program should not be changed (*join long-term security data property*).

The proposed protocol automaton model (Finite State Machine model, protocol FSM model) is defined as Moore automaton. Its states (and outputs) are the sets of binary transaction characteristics (zero for not shared data and one for already

shared). Input alphabet is a set of program names. Transition functions are defined by programs changing the state of the machine by sharing the values of certain transactions or removing shared ones i.e. data available for reading. We assume that there is an initialization protocol, one or more programs which execution transfers the protocol model from any state to the initial state (*the initial state automaton property*). Then, accordingly to the input word (sequence of the programs names) consistently with the current states of the automaton, programs of the protocol execute and change in a certain way the current states.

The state of protocol model can be viewed as a list of pairs (share.attribute, attribute_value) available for all parties (module "share").

The program line share.attribute=attribute_value maintains this list if the pair with the same first element absent, otherwise it updates this pair replacing it by the pair with the same first element and new attribut_value.

The program line del share.attribute deletes the list pair with the first element share.attribute if the pair with such first element exists.

The content of list defines the predicate hasattr(share, "attribute") value. It is true, if a pair with the first element share.attribute present in the list, otherwise it is false.

Conditions for execution of the program on the input signal with its name are specified by some Boolean function (or Boolean expression) on the set of the protocol states. Program or their fragments are performed if for the current state of the protocol corresponding function has the value of 1 (or Boolean expression gets the value true). Otherwise the program or fragment thereof will not be executed and the protocol state is not changed (*the finite state automaton property*).

Execution of the program under which the protocol state and the values of all shared transactions as well as the other data do not change will be called "idle." Under certain protocol states (i.e. when transaction share.corrupt_A with its name A was shared) the program is executed in a corrupt version. A program corrupted by default is a program of the intruder that could also corrupt other programs within certain limits, sharing relevant transactions. Sequence in which the protocol programs are executed (with possible repetitions or "idle" calls) is defined by the protocol trace. Trace of the protocol is a sequence of program names (including references to the program protocol input data), that is, the automaton input word. In general, the number of traces, as well as their length is not limited. There are shortest traces. They correspond to the standard description of protocol with or without the intruder in the absence of "idle" calls. Initial segment of the trace corresponds to the protocol initialization. The subsequent segments of the trace will be called as work segments. Upon completion of the execution of each work segment the protocol model turns out to the initial state or in the state in which the executions of all programs except initializing are idle, the latest version is related to single use protocols (*returnable recurrence of the initial state property*). In view of mutually distance of participants, not always the actual trace may be predicted in advance, in particular, it may not conform to the standard protocol trace definition. We assume that the random selection for the execution of the next program, that is, for random automaton input word after removing the names

of programs that are executed "idle", it turns out to be the initial segment of one of the shortest possible traces (traces *equivalence with shortest representatives property*).

Finally, we assume that in case of two runs of the same program, the second run should be idle (programs exhaustive execution property).

The properties of protocol models listed above should be provided by their component programs. Protocol models that possess this property will be called finite state automaton compositional (FSM compositional) models.

In the study of the protocols and attacks on them on the basis of the proposed automaton model, we believe further that there is a Supervisor, specifying the input word of the automaton model and representing the output of the protocol as output data.

Remarks. Functions of Supervisor can be performed by the operator defining the input by editing programs and calling protocol programs in a certain order.

Example 1. FSM model of the Diffie-Hellman protocol [8] is represented in Table 1. Parties: the trusted center program T, party A, and party B. Input: dscr EC(program T) is description of elliptic curve parameters; Output: two copies of secret key key=key_A(program A), key=key_B(program B). The protocol (and FSM) states are represented by binary vector

$$S= (\text{hasattr(share,"T")}, \text{hasattr(share,"A")}, \text{hasattr(share,"B")}) \in \{0,1\}^3.$$

The state diagram of FSM is depicted in Fig. 1. The shortest traces (of length 5): T(dscr EC),A,B,A,B; T(dscr EC),B,A,B,A; the trace T(dscr EC),A,*A*,B,*B*, *B*,A,*A*,B, *B* is equivalent to the first shortest trace.

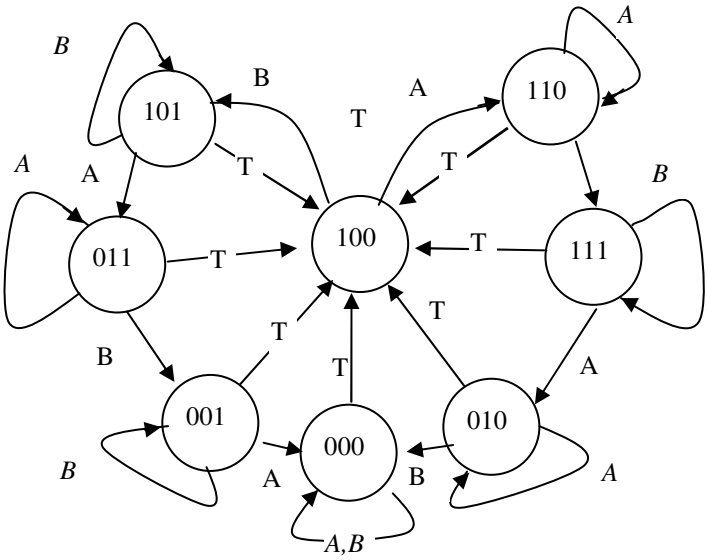


Fig. 1. The diagram of FSM model of Diffie-Hellman protocol. The idle executions of some programs are shown by loop arrows and are denoted in italic

Table 1. FSM compositional models of the Diffie-Hellman protocol (programs T, A and B) and of the Diffie-Hellman protocol attacked a man in the middle (programs Mini, T, A, B and M)

T, MIni	A	B	M

T:
INPUT:
description of elliptic curve EC
generate EC elliptic curve point P of large order N
share.T=(C,P)
del share.A
del share.B
 	
MIni:
 	
step=o
	if S=(1,1,1):
	key= a*B
	del share.B
	del share.T
	else:
	if S=(1,0,0) or S=(1,0,1):
	C=share.T [0]
	P=share.T[1]
	N= ord P
	$a \in_U (1,N)$
	A=a*P
	share.A
	del share.B
	if S=(0,0,1):
	key=a*B
	del share.B
		if S=(0,1,1):
		key=b*A
		del share.A
		else:
		if S=(1,0,0) or S=(1,1,0):
		C=share.T [0]
		P=share.T [1]
		N= ord P
		$b \in_U (1,N)$
		B=b*P
		share.B
		else:
		if S=(0,1,0):
		key=b*A
		del share.A
			if S=(1,0,0) and step=0:
			C=share.iniECGFp[0]
			P=share.iniECGFp[1]
			N= ord P
			$m \in_U (1,N)$
			M=mP
			step=step+1
			if S=(1,1,0) and step=0:
			aP=share.A
			key_{MA}=m*aP
			share.A=M
			step=step+1
			if S=(1,1,1) and step=2:
			bP=share.B
			key_{MB}=m*bP
			share.B=M
			step=0
			if S=(1,0,1) and step=1:
			bP=share.B
			key_{MB}=m*bP
			share.B=M
			step=step+1
			if S=(0,1,1) and step=2:
			aP=share.A
			key_{MA} =m*aP
			share.A=M
			step=0

Example 2. The ElGamal signature cryptosystem: illustration of FSM-protocol model *join long-term security data property*.

Table 2. FSM model of the ElGamal signature

T	P, S	V

del share.publickey
del share.signature
	if S=(0,0):
	generate *secret key a* and public key *public_key*
	share.publickey=*public_key*
	if S=(1,0):
	for given *message*
	compute (*message*,*r*,*s*)
	share.signature=(*message*,*r*,*s*)
		if S=(1,1):
		p=share.publickey[0]
		alpha=share.publickey[1]
		y=share.publickey[2]
		message=share.signature[0]
		r=share.signature[0]
		s=share.signature[1]
		return Verivy(*messag, r, s*)
		del share.signature

FSM model of the ElGamal signature cryptosystem is schematically represented in Table 2. The state diagram of FSM is depicted in Fig. 2. The protocol (and FSM) states are represented by binary vector S= (hasattr(share,"publickey"), hasattr(share,"signature"). Signing of the messages $M_1,\ldots, M_i,\ldots, M_n$ at the same once generated secret key *key* is performed according to the following trace $T,P,S(M_1),V,\ldots, P,S(M_i),V, \ldots, P,S(M_n),V$. The idle runs are shown in italic.

3 Composition of FSM Protocol Models

The properties of FSM compositional protocol models allow constructing their compositions accordingly the simplest rules. The composition $\pi(\pi_1,\ldots, \pi_t)$ of two or more compatible protocol models $\pi,\pi_1,\ldots, \pi_t$ is the protocol model included unified edition of models π_t, $i=1,\ldots,t$ and the master protocol model π controlling their interaction. Protocol models are compatible, if they allow the same description of underling algebraic structure and the same types and dimensions of elements in interconnecting (input- output) pairs.

Mentioned unified edition supposed that

a) The sets of the names of shared data generated by distinct protocol models are disjoint;
b) The initialized protocol models are combined into a single initialized protocol model eliminating duplication and initializing some algebraic structure parameters and primitives;
c) The master protocol corresponds to the desired interaction;
d) The composition should be FSM compositional.

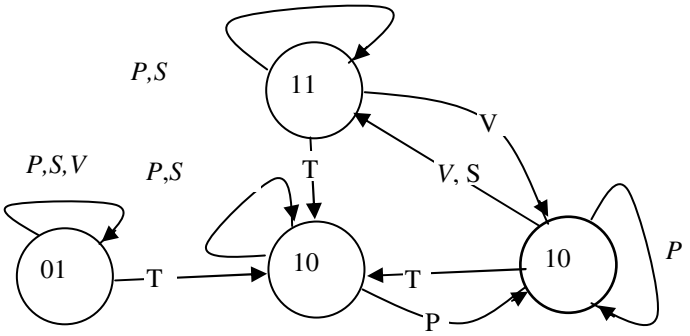


Fig. 2. The diagram of FSM model of GlGamal signature protocol

This approach to composition of protocol models allows simulation of protocol processing under intervention of intruder.

Protocol with the intruder is the one with another program M assigned to the adversary network party which operates on the same or expanded space of input data like the programs of the original protocol, but can perform the following functions:

- Interception of "shared" output data of the original protocol programs (canceling prefix «share.», while storing the values in computer);
- Replacement of values of "shared" output data;
- Generation of output data available to read (shared data);
- The corruption of the participants;

The aim of the adversary in this case is to obtain (as output of its program) certain private data of members of the network or to form the output data of the participants that are data-dependent of the attacker.

Example 3. Man-in-the-middle-attack on the Diffie–Hellman protocol. There are two programs A and B as well as the initialization program T. They are the same as in Example 1 and form the protocol π_1 FSM model represented in Table 1 and depicted in Fig.1. Additionally there is an intruder protocolπ model. It contains the initializing protocol model that consists of program T (the same as in Table 1) and the program MIni initializing the step number (see Table 1).

As well, it contains the program M represented in Table 1.

FSM model of composition $\pi(\pi_1)$ obtained by combining the FSM model of protocol in the Table 1 and intruders protocol without duplication of programs T. That is composition of Diffie-Hellman and adversary protocols $\pi(\pi_1)$={T, Mini, A, B, M}. Its transition table ***T*** is given in Table 4.

Table 2. Transition table **T** of Moore FSM $\pi(\pi_1)$={T, Mini, A, B, M}

T	000	001	010	011	100	101	110	111
T	100	100	100	100	100	100	100	100
A	000	000	010	011	110	011	110	010
B	000	001	000	011	101	101	111	111
M	000	001	010	011	100	101	110	111

The adversary succeeds if participates in the following traces:

T,Mini,M,A,M,B,M,A,B;

T,Mini,M,B,M,A,M,B,A.

If successful, the adversary has a shared key with one party and another shared key with second participant. In other traces its actions do not affect the results. Omitting the details describing the interaction of the program M with programs A and B note that parties A and B do not learn, whether the program M participated in this interaction effectively.

Example 3. Parallel session attack on Woo-Lan protocol [9]. Programs B,A, and T are the programs of two parties and of trusted center. M is the program of possibly corrupted party. There exists adversary that could implement the program Adv share.corruptM

Then it is easy to write the program M such that when M is not corrupted (Adv was not implemented) B initializes contacts with A and M roughly at the same time using Woo-Lan protocol. If M is corrupted (Adv was implemented) it can realize attack discovered by Abadi and Needman [10] blocking messages flowing to A. In the

corrupted version B believes that A is corresponding to it in a run while in fact A has not participated in the run at all [1]. One can simulate many contemporary systems such as pairing based cryptographic protocols [11,12,13] or privacy-preserving smart metering system [14]. Those FSM models where simulated using Algebraic processor presented in text section.

4 The Structure and Functionality of Algebraic Processor

In this section we present some complex (but realizable using algebraic libraries and python web functions) software called Algebraic processor. In particular it supported FSM simulation method proposed in this chapter providing algebraic calculations and maintaining data sharing between distinct parties. The main aim of the algebraic processor is to provide shared access to the interpreter for remote users. Each user is running his programs, either written by his own or loaded from the server on demand. The programs operate two types of data: private data and shared data in a "share" module which can be the result of other users' work. To synchronize the runs users are checking if the required attributes exist in that module. Adversary can delay the execution by removing an attribute (program line del share.attribute) and publish a replacement of an attribute.

To perform algebraic calculations we use the capabilities of python C API[15], which allows us to use functions from algebraic libraries and compiled modules within python program.

This approach integrates different software components (see Fig.2), providing access to functionality of Algebraic library C++ in the absence of its counterpart in the scripting language. Usage of SWIG[16] interface generator minimizes the work necessary to process the large amount of code. SWIG automatically generates accessor functions for specific classes and methods using the declaration of data type conversions given in the Interface modules. The Interpreter with the Library binding can then be compiled by Emscripten[17] (Emcc in Fig.2) giving the Javascript library. Emscripten is an LLVM-to-JavaScript compiler. It takes LLVM bitcode[18] - code representation that provides type safety, low-level operations, flexibility, and the capability of representing many high-level languages universally. It can be generated from C/C++, using Clang[19] compiler. Emcc compiles it into JavaScript, which allows users to execute programs right in their browsers. Though Javascript code is interpreted, modern JavaScript VMs are quite speedy, and even more so with asm.js – a low-level subset of JavaScript that helps JavaScript engines by making it simpler to detect types and avoid them changing later. Compiled code from a statically-typed language is still implicitly typed, so optimizable on the client side. Since public transactions only determine the data flow, one program can be used to perform a simulation of multi-party protocol execution on one server using just python with the library binding and for actual usage on the web with a server as shared storage [20].

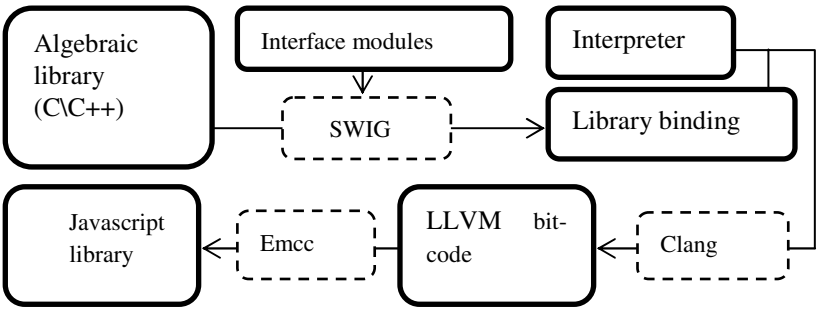


Fig. 3. Structure of algebraic processor building environment

5 Conclusions

In this chapter, a new approach to modeling cryptographic protocols based on finite state machine model that reflects both the system functionality and strategy of attacks has been proposed. Proposed models will allow essentially contract the traces space in automatic security protocol analysis. This approach is supported by created software that allows computer experiments to identify and demonstrate the security vulnerabilities. The effectiveness as an approach and processor has been confirmed by simulation of many famous cryptosystems. The algebraic processor is used in undergraduate education. Although we considered one session processes between the specified parties, currently multi-session version of the algebraic processor for modeling of parallel execution of multiple copies of the protocols has been developed and tested.

Acknowledgements. FSM model of privacy-preserving smart metering system was studied by Nikita Marin. This research was carried out with the financial support of the Russian Foundation for Basic Research, project 14-01-00671.

References

1. Mao, W.: Theory and Practice. Hewlett Packard Company. Prentice-Hall Inc. PTR, New Jersey (2004)
2. Cremers, C., Lafourcade, P.: Comparing State Spaces in Automatic Security Protocol Verification. Electronic Notes in Theoretical Computer Science. ETH Technical Report, No. 558 (2007), http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/CL-avocs07.pdf
3. Canetti, R.: Security and Composition of Cryptographic Protocols: A Tutorial. Technical Report 2006/465, Cryptology ePrint Archive (2006), http://eprint.iacr.org/2006/465
4. Kousters, R., Tuengerthal, M.: Joint State Composition Theorems for public-Key Encryption and Digital Signature Functionalities with Local Computation, http://eprint.iacr.org/2008/006.pdf

5. Raju, R., Shanmugapriya, S., Mahalakshmi, P., Lalitha, G.: Providing security for Web Service Composition using Finite State Machine. International Journal of Computers & Technology 4(2) (March-April 2013)
6. Jan Jürjens, J., Guido Wimmel, G.: Security Modelling for Electronic Commerce: The Common Electronic Purse Specifications, http://www4.informatik.tu-muenchen.de/publ/papers/JurWim00.pdf
7. Wason, R., Ahmed, P., Qasim Rafig, M.: Automata-Based Reliability Model: The key to Reliable Software. International Journal of Software Engineering and Its Applications 7(6), 111–126 (2013)
8. Dffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Info. Theory IT-22(6), 644–654 (1976)
9. Woo, T.Y.C., Lam, S.S.: Authentication for distributed systems. Computer 25(1), 39–52 (1992)
10. Abadi, M., Needman, R.: Prudent engineering practice for cryptographic protocols. Technocal Report DEC SRC Technical Peport 125. Digital Equipment Corporation (November 1995)
11. Joux, A.: One round protocol for tripartite Diffie-Hellman, LNCS. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 385–393. Springer, Heidelberg (2000)
12. Boneh, D., Lynn, B., Shacham, H.: Short signatures from Weil pairing. J. of Cryptology 7, 297–319 (2004)
13. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM J. on Computing 32, 586–615 (2003)
14. Rottondi, C., Verticale, G., Capone, A.: Privacy-preserving smart metering with multiple da ta consumers. Computer Networks 57, 1699–1713 (2013)
15. Python/C API Reference Manual, http://docs.python.org/2/c-api/
16. Simplified Wrapper and Interface Generator, http://www.swig.org/
17. Emscripten: An LLVM-to-JavaScript Compiler, https://github.com/kripken/emscripten
18. Low Level Virtual Machine Intermediate Representation Bitcode, http://llvm.org/docs/BitCodeFormat.html
19. Clang: a C language family frontend for LLVM, http://clang.llvm.org/
20. Remote educational resource MPEI Processor, http://mm.mpei.ac.ru:8080/eng/

Disturbance Injection in Dependability Assessment of Android Applications

Piotr Gawkowski and Maciej Sułek

Institute of Computer Science, Warsaw University of Technology (WUT)
Nowowiejska 15/19. 00-665 Warszawa, Poland
P.Gawkowski@ii.pw.edu.pl

Abstract. The paper presents a software technique of high level error emulation toward better test coverage of Android applications. It is targeted on error detection and handling procedures improving the manageability of the testing and deployment of software. The fault injection concept is successfully implemented in the fault injection tool that proved its usability in a set of experiments with some popular Android applications.

Keywords: fault injection, software testing, Android, Java, dependability, error handling, test coverage.

1 Introduction

Along with the growing popularity and functionality of mobile devices also users' expectations on software quality are increasing. However, mobile devices introduce more complex scenarios of possible disturbances in application operation than in the desktop computer environment. Multiplicity of system modules as well as usual problems with availability of system services (like geolocation or network range) or low levels of available resources (computing power, memory space, file system accessibility and performance, etc.) are common in the mobile systems. Nevertheless, users expect that even in such critical conditions the whole system and applications will behave correctly.

On that market of mobile devices the unquestioned leader at the moment is Android operating system. The first phone with Android was available on the market in 2008. The user, after a few "clicks" in the virtual store (now called Google Play) is able to download and install the desired application. Such distribution schema was greatly appreciated by the users as till January 2014 there are more than 1 million applications available in the store. More than 80% of them are for free. However, 22% is considered as low quality applications [1].

The practical weakness of classical testing techniques is poor coverage of the error and exception handling execution paths that are crucial from dependability perspective. They are responsible for handling failed system calls, unavailability of required resources and services, their failures, etc. Even if such events are rare, the user expects that the responsible handling code will take some reasonable actions. They might be, for instance, an adequate warning message for the user in the simplest case,

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_19,

saving users data processed at the critical instant, or a try to mitigate error effects and tolerate an error. Despite the complexity of the handling procedures, there is a practical problem from the developer and tester perspective how to verify and test such procedures as possibly critical circumstances (especially hardware related) may be very rare. That makes testing of such procedures a challenge.

Usually, in order to verify the error handling codes, developers artificially provoke necessary conditions within a system or application. In most cases it means a temporal source code changes within the target application to redirect the execution paths to the desired ones. Such manual fault injection has several drawbacks. First of all, it requires accessibility of the application or/and system/library source code. Problems arise if the dependability of a library or a service module with the closed code should be verified (e.g. to check how they behave in critical conditions). Secondly, manual changes of the target code are time consuming task that have to be followed by additional compilation. Introducing changes in the code there is also increased probability that some new bugs will affect the final code. Finally, in many companies such testing is considered as "waste of money" – the time-to-market pressure and problems with correct implementation of the mainstream application functionality eclipse the problem of error handling procedures testing to the margin.

Software implemented fault injection (SWIFI) approach promises solution in the considered problem. However, even if the SWIFI concepts are quite old ([2-5] and references therein) they were not warmly accepted in software companies. Only few companies can share a success stories on using fault injection in practice. SWIFI is still considered as academia related or exotic technique for verification of high reliability demanding systems rather than a practical approach that can be helpful in everyday software development. The low interest of the companies on SWIFI is somehow understandable. There are only few out-of-the-box solutions (see Sect. 3) on the market. Moreover, applying them in practice requires a lot of effort to organise experiments and, without prior experience, the obtained results could be misleading or simply useless in software enhancement. Recently the interest on fault injection seems to be growing. Fault injection framework (with very limited functionality) exist for instance in the Linux kernel and was used to evaluate distributed systems. Another practical example, related to the main subject of the paper, is LRFI system [6]. Some other tools, developed in the authors' Institute of Computer Science WUT, are described in [7]. This paper deals with the SWIFI based application testing under Android. The new fault injection system dedicated to application testing directly on Android device is presented along with summary of some experiments conducted on a set of well-known applications.

The paper is organized as follow. Section 2 introduces the software fault injection concept. Some fault injection tools are presented in Sect. 3. The Android Fault Injection System (AFIS) implementation and its main functionality is described in Sect. 4. Some practical experiments illustrating suitability of the developed system are reported in Sect. 5. The paper concludes in Sect. 6.

2 Software Implemented Fault Injection

The concept of the fault injection is based on the specific states enforcement (e.g. corrupted or unusual) in the system under tests (SUT). The new system state can be related to simple fault effect (e.g. bit flip in system memory cell) or sophisticated scenario of erroneous behavior of the target subsystem, device or software component. After the injection the SUT is examined towards any possible anomalies. Some of them can be severe – others might be acceptable from the end-user perspective.

Organizing fault injection experiments several aspects have to be considered. Different fault injection techniques and fault models should be used depending on the goal of the experiment. It can be targeted on characterization of overall system dependability, identification of the weakest system components or severe system responses, verification of the fault/error handling mechanisms effectiveness. The goal determines the fault model, the technique of injection and required controllability over the injection process and effects observability.

Hardware faults can verify the robustness of the system in the aggressive (e.g. environmental) conditions – physically provoked faults are more representative then in the other fault injection techniques in this case. However, due to limited controllability and observability of fault effects, system designer will probably get only little feedback to let him harden the design, especially in terms of its software.

Much more flexible are software implemented fault injection (SWIFI) approaches. Here no additional hardware is necessary to carry out experiments and more sophisticated fault models are possible to be implemented. The controllability and observability is much better than in case of hardware implemented fault injection. Basically, in SWIFI, the faults existence is rather emulated through their effects in software (i.e. error). In fact, many fault injection systems should be called *error injectors* but the *fault injector* term is commonly used in the community. It is worth noting that the SWIFI concept is related to the way the faults are introduced in the system but not the fault model it actually implements. Most of the research focuses on the analysis of temporary hardware faults – usually limited to the single bit-flip model [2], [3]. Such research is indisputably valuable; nevertheless, these tools are almost impractical from the standard software designer, developer or tester point of view.

There are also some works on the analysis of the effects of typical software bugs (i.e. originated in the software development) in the literature. Some of them are using mutation testing approach to estimate the number of yet uncovered bugs in the software and evaluate the set of the tests to maximize the test set coverage over the source code [4],[5],[8]. For some reasons this concept is also not much popular in the software companies as it requires a lot of effort that do not assure much better software quality in the end (i.e. is covering considered types of bugs really improves software quality?). At the other hand, erroneous calls and exception handling procedures are not covered by classical software testing approaches. In [2] the authors claim that such procedures can constitute even 2/3 of the overall source code. At the same time, the end-users, expect that the software will always behave in rational way (will allow saving the document, other tasks will not be disrupted etc.). The problem is to provide

a tester an easy-to-use system that could transparently emulate a representative set of real-life conditions in the SUT.

SWIFI approach can cover the described above problem. In some sophisticated scenarios the injected faults may emulate incorrect behavior of hardware devices as well as software components (e.g. delayed I/O responses, device error notifications, services failures, device or service unavailability, corrupted communication data, or thrown exception). In more general, emulating some circumstances through the disturbance injection (otherwise hard to arrange) during the execution of the system/application under tests (SUT/AUT) activates software parts that are never covered with traditional testing.

Moreover, to support the developer with valuable feedback data from the experiments, the fault injection system should have the ability to select and identify the AUT within the SUT. The disturbances introduced by the SWIFI could be limited to the AUT's context if necessary. The tool functionality and disturbance model should be understandable in order to be used by application users and testers to effectively verify the application behaviour in some unusual circumstances. Preferably, the result of such evaluation should provide detailed information allowing to trace down the reasons of the observed effects and locate parts of the application code which are responsible for these effects. Some effects are hard to be recognized in software – human interaction can be irreplaceable. The categorisation of observed effects (based on collected SUT and AUT logs) can be valuably extended if, for instance, personal notes and subjective user opinion on the observed behaviour would be also collected.

To evaluate the dependability of any application within the SUT, the tool should support examination without the necessity of AUT source code availability. So, disturbances to be emulated should be injected at the layer between the application and operating system. Here, an important issue is the proper selection of disturbances model, i.e. at which interfaces disturbances have to be emulated and how to emulate these disturbances. In order to mimic the existence of a specific disturbance for a given application it has to be assured that the overall state of the system is not affected (e.g. failing request should not actually take place at the operating system - only the requesting application should be noticed about the emulated failure).

Another important aspect is the ability to observe how the AUT operates at the specific API (what system/library functions are used, how many calls are made, etc.). Profiling AUT before the actual testing is crucial to optimize the whole process. Testing an application without its source code the one will not know what modules, libraries and system calls the application uses. Without that knowledge the scope of the disturbances to be emulated during the experiment cannot be narrowed down. Moreover, the AUT scenario (e.g. data to be processed) can be tuned to maximize the utilization on specific interfaces (at which the disturbances are emulated).

3 Disturbance Oriented Tools

Most of the available disturbance emulating tools are dedicated to some specific targets (e.g. to verify HDFS system behaviour on some failure scenarios like node crash,

disk or network failure [9]). Others are general but hard to be used for programmers and testers. In the follow only some Android and Java related tools will be discussed.

Jaca uses the reflection mechanism to operate on the Java byte-code level [10]. With this mechanism it can corrupt methods' arguments, class attributes and methods' return values. Unfortunately, Jaca cannot emulate more sophisticated scenarios. Such functionality is offered in JInjector system developed in Institute of Computer Science WUT [7]. The injection technique used in the JInjector is based on the Aspect Oriented Programming approach (AOP) that addresses the so-called crosscutting concerns. In AOP such code can be defined once and "injected" automatically into all corresponding places. The code that realizes the crosscutting concern is called an advice while the expression defining all the target points for that advice is a pointcut. The actual places in code that fulfil the pointcut are called joinpoints (e.g. pointcut `call(public * java.sql.DriverManager.getConnection(..))` selects all calls of the public method `getConnection` of every object of class `DriverManager`). As a result, JInjector dynamically injects a user-defined code that can block the execution of the target API call, change its parameters values, delay execution, throw exception, modify the result of the given method, etc. Thus, complex disturbance scenarios are possible to be evaluated. Unfortunately, JInjector uses the AspectWerkz framework which is not developed anymore (the only one at the moment of creation that supported dynamic aspect installation at target application runtime without the need of AUT source code availability). So, right now its lifetime and applicability is limited.

In Android application development several tools towards application testing are usually used. Some of them are Java oriented (e.g. JUnit, Sureassert, FindBugs), while some are dedicated for Android. One of them is Android test framework [11]. Basically it is a library that inherits some mechanisms from JUnit package. Thus, it is oriented on automated testing of some properties and assertions (e.g. if the object returned exists verify if the value is equal to the expected one). There are two main weaknesses of this framework: the tester/programmer has to identify the points of assertions and there is no possibility to emulate any unexpected environmental or possibly critical situations during AUT execution. Similarly, the Android Mock library can be used to create an object imitating the tested classes and interfaces. Such object can execute a test scenario based on the verification of the obtained results with the expected ones. Another tool, Robolectric [12], has also similar goal, however, through the annotations, it easy to steer some environment and system parameters (e.g. screen resolution, localization, API version).

None of the presented tools resolves the problems noted in Section 2 as there is no functionality of provoking failing system/library calls, device or service unavailability, etc. So, the parts of the code responsible for handling these situations are not covered with such testing. Moreover, Android is dedicated to mobile systems. Applications are using some specific modules like accelerometers, gyroscopes, GPS, etc. So, the probability of errors and device/service unavailability during system operation is much higher than in case of traditional computer systems and more complex to be

simulated at runtime. Development of such applications requires cross compilation. All of that make testing of such applications towards error handling a challenge.

4 AFIS – Android Application Testing

Android is based on Linux kernel but the basic language for user application developers is Java. However, applications are executed using a specially developed virtual machine called Dalvik. All the functionality needed for the applications is provided through the set of libraries and classes, e.g.:

- Reading data from device sensors (accelerometer, compass, light, etc.) with *SensorManager* class,
- Phone management (dialing and calls registration, etc.) – *TelephonyManager* class,
- Views management through the *SystemView* and *WindowManager* classes,
- Content delivery and data management (e.g. contacts, SMS) through *ContentProvider* class,
- Notification area management with *NotificationManager* class,
- Localization provider – *LocationManager* class.

Applications using these classes should provide proper handles in case of exceptions or simply failed method calls. At the same time, the system and libraries should be proofed against malicious parameters, improper sequence of methods calls, etc.

Developed Android Fault Injection System (AFIS) targets on the assessment of critical situations and errors handling by the applications directly on the target Android device. It emulates the disturbances by overloading methods of the open-source Android libraries. That requires modification of the original source code of these libraries in order to allow dynamic overloading of these methods at runtime. It is worth noting that the code of the overloaded method can be edited by the experimenter just before the experiment (i.e. the disturbance related to the specific method can be easily customized). Unfortunately, not every Android library is open-sourced. Because of the license limitations some standard Java libraries are not possible to be covered with the proposed solution. Nevertheless, currently, AFIS allows customized overloading of 125 methods of Android.

The main drawback of AFIS is the necessity to replace the original firmware of the target device with the instrumented one. However, the whole process of instrumentation is automated. Along with AFIS the modification script is provided. It takes the list of methods to be modified from the csv file. After the instrumentation step, the Android system should be compiled and can be applied to the target SUT. Currently, all the devices supported by the Google can be used (all from the Nexus family, HTC One, Samsung Galaxy S4 and Sony Xperia Z) as well as emulator on the PC.

Introduced modifications allow four scenarios of execution: 1) target method execution is registered for the profiling proposes (disturbances are not introduced), 2) the overloaded method is called instead of the target one, 3) the parameters of the target method call are disturbed, 4) the target method is called normally if the calling

application is different than AUT. In AFIS, four main components can be distinguished, two located on the desktop PC and two on the target Android device:

- AFIS Core – it contains modifications of Android source code that enables dynamic code injections to libraries' methods. It is compiled along with the whole Android system.
- AFIS Desktop Manager – it allows target system management from the desktop PC (automatic generation and compilation of overloaded methods to be injected to the target system, modification of some global parameters, up/down-loading files to/from the target system, setting the configuration for the tests, etc.).
- AFIS Desktop Analyzer – it collects experiments results. A main part of it is system log analyzer allowing inter alia creation of the AUT call graph.
- AFIS Android Manager – Android application running on the SUT that can be used as a standalone experiment manager (with some limitations to the full AFIS configuration).

Before the actual experiment the profiling execution of the AUT should be made (undisturbed). Through the logs collected during this step the AFIS Desktop Analyzer creates a call graph which shows interfaces used by the AUT (how many times and from which places the given method is called). After this step, the marginal and unused methods can be eliminated from further analysis. Using the Desktop Manager the configuration for the tests is set: the target method (to be overloaded) is generated, modified (to emulate disturbance) and compiled. The overloading method along with the tests configuration is then sent to the SUT. At this time, based on the prepared configuration, the AFIS will start to disturb the given AUT. If the modified methods are already stored on the target device, the experiment can be managed on the SUT itself with the AFIS Android Manager application (a screenshots of which is presented in Fig. 1).

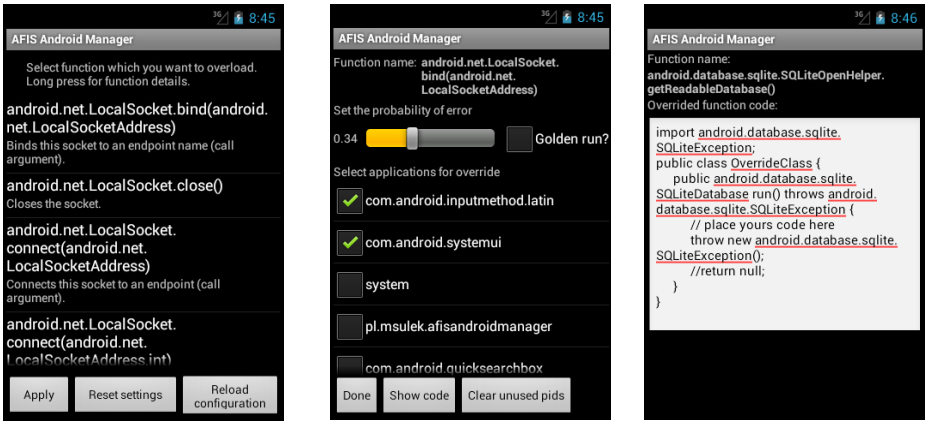


Fig. 1. AFIS Android Manager screenshots (from the left: selection of methods to disturb, selection of target applications, overview of the disturbing code)

In the research we concentrated on evaluation how the failing API methods influence the target applications – if the error handling procedures are adequate to the critical situation. So, the disturbance model was to emulate possible erroneous executions of the system API calls. In Java the main way of error notification is exception throwing. So, in the most cases the disturbances were emulated with throwing a proper exception while AUT operates with multimedia, databases, accesses to the memory cards, geolocating the device, etc. After disturbance injection the AUT behaviour was observed (subjectively by the user as well as through the analysis of system and application logs) and each test was categorized into one of the following classes of disturbance effects:

1. Everything executed without any noticeable side-effect,
2. All observable side effects were acceptable,
3. Some observable, non-critical but unacceptable side-effects were present but the AUT correctly notified the user upon that,
4. Some critical side-effects were correctly reported to the user,
5. Critical side-effects were not reported to the user,
6. Application was closed or terminated,
7. Abnormal operation was made (not forced by the AFIS itself).

Generally, the list above should be changed to distinguish different aspects - AUT specific (compare the list with [6]). It is worth noting, that the AFIS can collect many information upon the emulated disturbance (e.g. from which point in the AUT the disturbed API call was made, disturbance parameters). That provides a valuable feedback to the application developer: in case of application misbehaviour the one can point out the exact place in code. Moreover, the observed behaviour can be subjectively commented by the user after each AUT execution giving the basis to extend the disturbance effects set to new classes of criticality.

5 Practical Experiments

Using the AFIS (Section 4) some well-known applications were tested (more than 1M installations each): Connectbot (SSH client), Endomondo (personal sport tracking application), and VLC (multimedia player). Experiments were preceded with the profiling executions, so only used API methods were tested in experiments.

In case of the Connectbot application any error in `getWritableDatabase()` and `getReadableDatabase()` methods causes critical error or application shutdown. Additionally, any disturbance during the application startup causes its immediate shutdown. Generally, 76% of disturbances provoked application shutdown, 23% some critical effects without any notifications and only less than 1% of tests run smoothly without any noticeable impact.

One of the main features of the Endomondo application is the ability to collect the details of personal training along with the paths of, for instance, jogging on the map.

So, the application uses geolocation. Unfortunately, the experiments showed that errors in these service are not correctly handled. They cause application shutdown or lack of any reactions to the disturbance. In case of this application more complex behavior scenarios were observed (see Fig. 2).

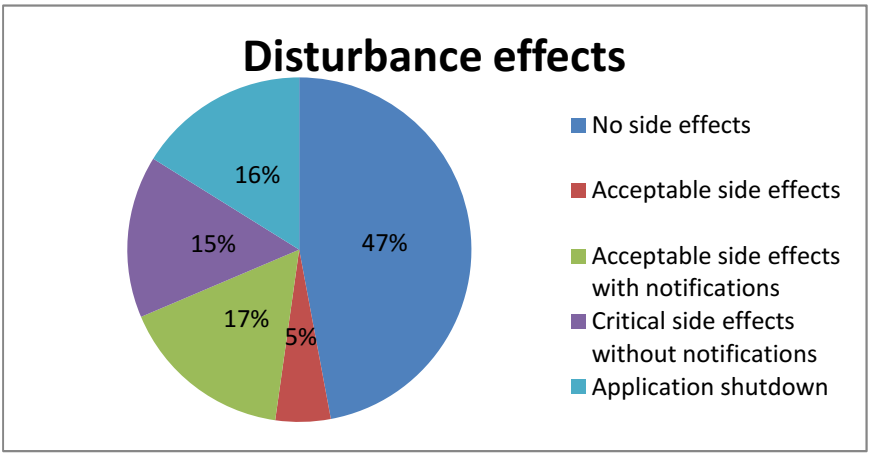


Fig. 2. Summary of Endomondo experimental results

In case of VLC multimedia player the most critical error found was incorrect handling of the method checking the availability of the memory card. In the considered scenario, there was a video file in the VLC's playlist that was stored on external memory card. Before accessing that card VLC checks its availability with the call to `Environment.getExternalStorageState()` method. The disturbance was configured to report the unavailability of the card to the calling application. In such case the application should not try to operate on the not existing device. Apparently, VLC ignores the reported state as VLC was trying to access the file from the playlist. Moreover, the unavailability of the application's database always blocked the application to start up.

These few examples show that the developed AFIS can be very useful in finding out software quality issues. A set of predefined erroneous scenarios and environments can be prepared according to some dependability policies to allow target software verification. In effect, the test coverage can be substantially improved with SWIFI. The proper fault model can result in valuable feedback for software developers facilitating management tasks and code quality improvements. However that requires some extra work. Nevertheless, the developed disturbances can be reused and shared between companies and projects. One of the main AFIS drawback is the necessity to instrument the target device firmware. That requires the accessibility of the source code, libraries, drivers etc., compilation and reprogramming the device. Moreover, the experiments are mostly valuable when the disturbance analysis is deep – in reality it means the necessity of analysis of several logs, user notes etc.

6 Conclusion

Classical fault injection is not much popular in the software developing community as the knowledge gained from experiments is unlikely helpful in solving typical problems of testers and developers. However, moving the injected fault model to the levels of disturbances and erroneous API calls changes this shallow reputation. In the paper a not traditional technique of software implemented disturbance injection is presented. It is targeted at checking the reaction of the AUT to some unusual but realistic circumstances like I/O errors, low resources, network errors, etc. The goal is to allow software developers and testers checking the correctness of AUT behavior in such situations. Some experiments reported in the paper show that applications can be easily improved at least by proper notification of the erroneous conditions to the end-user.

Such disturbance injection tools can also support quality verification of the third party software libraries and modules against dependability properties (like reliability, security, safety, stability etc.). Developed AFIS system can be easily adopted to different target devices. The ability to conduct experiments without the need of complicated configuration should be noted. That extends the classical SWIFI concept with quite different goals and fault models (here related to disturbances or errors). They are more practical for software developers and testers as they exist in conceptual level of software.

References

1. Android statistics webpage (January 26, 2014), http://www.appbrain.com/stats/
2. Lyu, M.R. (ed.): Software Fault Tolerance. John Wiley, Chichester (1995)
3. Ziade, H., Ayoubi, R., Velazco, R.: A Survey on Fault Injection Techniques. The Int'l Arab Journal of Inf. Technology 1(2), 171–186 (2004)
4. Madeira, H., Costa, D., Vieira, M.: On the emulation of software faults by software fault injection. In: IEEE Int'l Conf. on Depend. Systems and Networks, pp. 417–426 (2000)
5. Bieman, J.M., Dreilinger, D., Lin, L.: Using fault injection to increase software test coverage. In: Proc. of the 7th Int'l Symp. on Software Reliability Engineering, pp. 166–174. IEEE Computer Society, Washington (1996)
6. Gawkowski, P., Pawełczyk, P., Sosnowski, J., Cabaj, K., Gajda, M.: LRFI – Fault Injection Tool for Testing Mobile Software. In: Ryżko, D., Rybiński, H., Gawrysiak, P., Kryszkiewicz, M., et al. (eds.) Emerging Intelligent Technologies in Industry. SCI, vol. 369, pp. 269–282. Springer, Heidelberg (2011)
7. Gawkowski, P., Markowski, M., Smulko, G., et al.: Fault Injection Techniques Towards Software Quality Assessment. In: Jałowiecki, P., Łukasiewicz, P., Orłowski, A. (eds.) Information Systems in Management XVI: Modern ICT for Evaluation of Business Information Systems, pp. 17–28. SGGW, Warszawa (2012)
8. Ammann, P., Offutt, P.: Introduction to Software Testing. Cambridge University Press (2008)

9. Pallavi, J., Haryadi, S.G., Koushik, S.: PREFAIL: a programmable tool for multiple-failure injection. In: Proc. of the 2011 ACM Int'l Conf. on Object Oriented Programming Systems Languages and Applications, pp. 171–188. ACM, New York (2011)
10. JACA (October 2012), http://www.ic.unicamp.br/~eliane/JACA.html
11. Vogel, L.: Android application testing with the Android test framework (2013), http://www.vogella.com/articles/AndroidTesting/article.html
12. Robolectric: Unit Test your Android Application (September 12, 2013), http://pivotal.github.io/robolectric/

Approximate Algorithm for Fast Capacity Provisioning in WANs with Trade-Off between Performance and Cost under Budget Constraint

Mariusz Gola and Adam Czubak

Institute of Mathematics and Informatics, Opole University
ul. Oleska 48, 45-052 Opole, Poland
{mariusz.gola,adam.czubak}@math.uni.opole.pl

Abstract. Due to the emergence of Software Defined Networking (SDN) with the idea of centralized control over computer networks, the Capacity and Flow Assignment Problem (CFA) may be approached in a classical non-distributed fashion in real-life scenarios. The question arises whether a heuristical approach to this NP-complete problem is of any use in practice.

This paper is focused on the problem of configuring a Wide Area Network topology with trade-off between link cost and response time to users under budget constraint. The link capacities in the network and the routes used by packets are determined in a way to minimize network link cost and response time at the same time. Budget constraint means, that network link cost should be lower than the given budget value. An exact algorithm for CFA, due to its NP-completeness, is either not feasible or works well for small networks only. In this paper we propose an alternative heuristic solution, compare its effectiveness to the exact solution and finally show its running time when used for a real-life computer network. The experiments performed using standard off-the-shelf computer gear show that this solution is fast enough and surprisingly accurate and thus ready for implementation for an SDN controller.

Keywords: wide area networks, routing, capacity provisioning, SDN.

1 Introduction

The concept of Software Defined Networking is basically centralized network management and separation between the data and control planes [1], [13]. This simple idea of centralized control was considered in the past but abandoned and network designers turned to distributed algorithms or static configuration. The reason was the presumption that with a centralized approach a single point of failure follows. Contemporary network devices provide many technologies to tackle the above concern, like:

- Redundant power supplies;
- Virtualized operating systems, ie. two operating systems running in parallel in an active-standby fashion (ie. CISCO IOS XR);

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_20,

- Redundant network interface cards;
- Redundant uplink technology (ie. MLAG - Multichassis Link Aggregation).

All the above make nowadays a centralized controller-based network management system like SDN an appealing solution. Technologies like OpenFlow [11] or Netconf/XML are available implementations of this idea. Turn from distributed to centralized control over the network allow for the use of classical graph-based algorithms like exact (CFA) algorithms. It is worth to mention that an SDN controller has complete knowledge of the network including, but not limited to: nodes, topology (ie. links), link capacities. So the controller seems to be the right place in the network to calculate routes, flows and provision bandwidth with the constraints imposed by the link capacities. But due to computational complexity of CFA it is not feasible to run an implementation of the exact algorithm on the SDN controller.

Contemporary WANs are heterogeneous in nature regarding different layer 2 technologies available. Due to this, it is advisable to use a more abstract approach to this well-known computational problem as the results will be of use not only now but also when different future L2 technologies emerge in near future.

The problems of joint optimization of link capacities and flow assignment (routing) is called Capacity and Flow Assignment Problem (CFA). Several different formulations of this problem can be found in the literature. Generally, they correspond to different choices of performance measures, design variables, constraints and different types of flows [8], [12], [15], [17]. Exact algorithms for CFA problem are presented in the papers [8], [9], [12], [15]. Exact algorithm for the special case of CFA, where initial capacities are already installed on each link of the network is presented in [4]. In literature a heuristic approaches which is less exact but significantly faster for solving these problems could be found as well. A tabu search algorithm for the routing and capacity assignment problem in computer networks in [16] was also applied. Some other approaches to CFA problem based on genetic algorithms are presented in [7], [18], [14]. Some aspect of top down approach to TCFA (Topology Capacity Flows Assignment) problem was presented in [10].

In this paper as a criterion function we chose a function that considers both: network cost and network performance. Network performance is represented by average delay per packet. The routing problem is perceived as the multi-commodity flow problem [13]. To present the criterion function in the unified way, the following two distinct types of costs are considered:

- leasing cost: cost of channel used in the network;
- delay cost: total average delay per packet times unit cost of the delay.

The introduced criterion is the sum of the leasing cost and the delay cost and it is called the combined cost. Thus, the optimization problem with the combined cost criterion is formulated as follows:

given: *network topology, external traffic requirements, discrete cost-capacity function, budget of the wide area computer network*

minimize: *combined cost*

over: *channel capacities, routing (i.e. multicommodity flow)*

subject to: *multicommodity flow constraints, channel capacities constraints, budget constraint*

Channels' capacities constraints mean that capacity for channels could be chosen only from the set of discrete values of capacities. In this case the considered topology design problem is NP-complete [2].

2 Problem Formulation

Consider a WAN with n nodes and b channels which may be used to build the network. For each channel i there is the set $ZC^i = \{c_1^i, c_2^i, ..., c_{e(i)}^i\}$ of alternative capacities from which exactly one must be chosen. $e(i)$ is the number of capacities available for channel i. Let d_k^i be the cost of the leasing capacity value c_k^i for channel i [$/month]. Let capacities from each set ZC^i be ordered in such way that

$$c_1^i \geq c_2^i \geq ... \geq c_{e(i)}^i \tag{1}$$

Let x_k^i be the decision variable, which is equal to one if capacity c_k^i is assigned to channel i and x_k^i is equal to zero otherwise. Let $W^i = \{x_1^i, ..., x_{e(i)}^i\}$ be the set of all variables x_k^i which correspond to the channel i. Exactly one capacity from the set ZC^i must be chosen for channel i, then

$$\sum_{k=1}^{e(i)} x_k^i = 1 \quad \text{for} \quad i = 1,...,b \tag{2}$$

Let $W = \bigcup_{i=1}^{b} W^i$ be the permutation of values of all variables $x_k^i,\ k = 1,...,e(i),\ i = 1,...,b$ for which the condition (2) is satisfied, and let X_r be set of variables which are equal to one in W. The set X_r is called a selection. Then, each selection determines the unique values of channel capacities in the wide area computer network. Let S be the set of all possible selections of X_r.

Let $T(X_r)$ be the minimal average delay per packet [s/packet] in the wide area computer network in which values of channel capacities are given by the selection X_r. The expression on an average delay per packet is given by Kleinrock's formula [5]. $T(X_r)$ could be obtained by solving a multicommodity flow problem in the network with the channel capacities given by X_r [12], [9]. If this problem has

no solution for a given requirement matrix and for the given selection X_r, then $T(X_r) = \infty$.

Let $d(X_r)$ be the sum of leasing cost of capacities given by selection X_r, i.e.

$$d(X_r) = \sum_{x_k^i \in X_r} x_k^i d_k^i \tag{3}$$

Then, the objective function consisting of two distinct types of costs (delay cost and leasing cost) is following:

$$Q(X_r) = \alpha T(X_r) + d(X_r) \tag{4}$$

where α is the unit cost of delay [$/month/packet/s].
The CFA problem with the combined cost criterion function can be formulated as follows:

$$\min_{X_r} Q(X_r) \tag{5}$$

subject to

$$X_r \in S \tag{6}$$

$$d(X_r) \le B \tag{7}$$

where *B* denotes the budget (maximal feasible leasing capacity cost) of the wide area computer network. In [9] exact algorithm for solving above problem was presented. Impact of coefficient α on solution of problem (5-7) was presented in [3].

3 Calculation Scheme

Let $X_1 = \{x_1^1, x_1^2, ..., x_1^b\}$ be the initial selection. According to expression (1) the representation X_1 consists of channels set to maximal possible capacities. The variable x_k^i is called normal if $x_k^i \in X_r$. The remaining variables are called reverse. A replacement of any normal variable x_k^i by reverse variable from W^i is called complementing. Each selection X_{r+1} is obtained from a selection X_r by complementing one variable x_k^i by another variable x_j^i from the same set W^i.

Theorem 1.

Let $X_r \in S$. If the selection X_s is obtained from X_r by complementing the normal variable $x_k^i \in X_r$ by the reverse variable $x_j^i \in X_s$ then

$$Q(X_s) \le Q(X_r) - \Delta_{kj}^{ir} \tag{8}$$

where

$$\Delta_{kj}^{ir} = \begin{cases} \dfrac{\alpha}{\gamma}\left(\dfrac{f_{ir}}{c_k^i - f_{ir}} - \dfrac{f_{ir}}{c_j^i - f_{ir}}\right) + \left(d_k^i - d_j^i\right) & \text{if } c_j^i \ge f_{ir} \\ \dfrac{\alpha}{\gamma}\left(\dfrac{f_{ir}}{c_k^i - f_{ir}} - \dfrac{c_{\max}^{ir}}{\varepsilon}\right) + \left(d_k^i - d_j^i\right) & \text{otherwise} \end{cases} \tag{9}$$

and f_{ir} is equal to the optimum value of f_i which is obtained by solving the multicommodity flow problem for values of channel capacities given by X_r, and ε is the minimal feasible difference between capacity and flow in each channel $(0 < \varepsilon \le 1)$.

Proof.

In the case when $c_j^i > f_{ir}$, it is easy to show that the optimal multicommodity flow for the network with channel capacities given by X_r is also the feasible multicommodity flow for the network with channel capacities given by X_s. Let $T(X_s, f_r)$ be the average delay per packet calculated for the multicommodity flow $\underline{f}_r$. Since the selection X_s differs from X_r only with one variable, then

$$T(X_s, f_r) = T(X_r) - \frac{1}{\gamma}\left(\frac{f_{ir}}{c_k^i - f_{ir}} - \frac{f_{ir}}{c_j^i - f_{ir}}\right) \tag{10}$$

Since $T(X_s)$ is the minimum average delay per packet and $T(X_s) \le T(X_s, f_r)$, Moreover, $d(X_s) - d(X_r) = d_j^i - d_k^i$ and $\alpha > 0$. Then,

$$Q(X_s) = \alpha T(X_s) + d(X_s) \le \alpha T(X_s, f_r) + d(X_r) + d(X_s) - d(X_r) = Q(X_r) - \Delta_{kj}^{ir}.$$

In the case when $c_j^i < f_{ir}$, the expression $f_{is}/(c_j^i - f_{is})$ in $Q(X_s) - Q(X_r)$ may be upper bounded by $c_{\max}^{ir}/\varepsilon$ because $f_{is} < c_{\max}^{is} \le c_{\max}^{ir}$ and $c_j^i - f_{is} \ge \varepsilon$ ■

It follows from the Theorem 1 that we should choose for complementing such variables $x_k^i \in X_r$ and $x_j^i \in X_s$ for which the value of (9) is maximal and condition (6) is satisfied.

The algorithm goes through two phases:

1. In the first phase we start with representation X_1. We try to find a solution that complies with (7) and is a solution to the optimization problem (5-7). In the first

phase we start with representation X_1 with a maximal cost and we try to find a different solution fulfilling (7). For complementation we take variables x_j^i which correspond to opposite arcs (with lower capacities and thus lower costs). Every succeeding representation generated by the algorithm has a lower cost $d(X_r)$. The first phase ends when a representation X_r such that it fulfills (7) is found. If no such representation is found, the algorithm ends and it follows that no solution to (5-7) exists.

2. In the second phase we search throughout the neighborhood of the representation X_r found in the first phase with the premise that a better solution is located nearby. So we search in the neighborhood for a solution with lower Q, still imposing (7) on the results.

4 Heuristic Algorithm

Now we introduce an approximate algorithm used to solve the optimization problem (5-7). M is the set of reverse variable. If $M = \varnothing$ then we have no variables for complementing. K is the set of fixed normal variables; F is the set of fixed reverse variables. Fixed variables cannot be used for complementing for each successor X_r. W is the permutation of values of all variables and was introduced in problem formulation section.

Phase 1: finding feasible solution for problem

Step 1: Perform $X_1 = \{x_1^1, x_1^2, ..., x_1^b\}$, compute $d(X_1)$, $Q(X_1)$.
If $Q(X_1) = \infty$ then no solution for optimization problem (5-7) exists, stop algorithm otherwise go to step 2.

Step 2: If $d(X_1) \le B$ then perform $Q^* = Q(X_1)$ go to step 6 (phase 2)
otherwise, perform $F = \varnothing$, $K = \varnothing$, r=1, go to step 3.

Step 3: Perform $M = W - X_r - F$.
If $M = \varnothing$ then no solution for optimization problem (5-7) exists, stop algorithm otherwise choose the variable $x_k^i \in X_r$ and variable $w_j^i \in M$ for which the value Δ_{kj}^{ir} is maximal. Next generate the selection X_{r+1} by complementing x_k^i by x_j^i by letting $X_{r+1} = (X_r - \{x_k^i\}) \cup \{x_j^i\}$, go to step 4.

Step 4: Compute $Q(X_{r+1})$.
If $Q(X_{r+1}) = \infty$ then perform $F = F \cup \bigcup_{t=j}^{t \le e(i)} \{x_t^i\}$, then go to step 3

otherwise go to step 5.

Step 5: Perform r=r+1. Compute $d(X_r)$.

If $d(X_r) \le B$ then perform $Q^* = Q(X_r)$, $F = \varnothing$, go to step 6 (phase 2)

otherwise perform $F = F \cup \bigcup_{t=k}^{t<j} \{x_t^i\}$, go to step 3.

Phase 2

Step 6: Perform $df = B - d(X_r)$, $E = X_r - K$, $M^t = W - X_r - F$

$$M = \bigcup_{x_j^i \in M^t} \{x_j^i\} : (\{x_j^i\} - \{x_k^i\}) \le df.$$

If $E = \varnothing$ or $M = \varnothing$ then go to step 8

otherwise choose the variable $x_k^i \in E$ and variable $w_j^i \in M$ for which the value Δ_{kj}^{ir} is maximal. Next generate the selection X_{r+1} by complementing x_k^i by x_j^i i.e. by letting $X_{r+1} = (X_r - \{x_k^i\}) \cup \{x_j^i\}$. Perform $F = F \cup \{x_j^i\}$, go to step 7.

Step 7: Compute $Q(X_{r+1})$.

If $Q(X_{r+1}) \ge Q^*$ then perform $K = K \cup \{x_k^i\}$ go to step 6

otherwise $Q^* = Q(X_r)$, r=r+1, go to step 6.

Step 8: X_r is the best approximate solution for problem (5-7).

The overall computational complexity of above algorithm is equal to $b \cdot (e_{\max} - 1) + 1$, where $e_{\max}$ is maximal value of $e(i)$ for i=1...b.

5 Computational Results

The presented approximate algorithm was implemented in C++ code. Extensive numerical experiments have been performed with this algorithm for many different network topologies, different set of possible capacities for each of the channels and different workload. The experiments were conducted to examine the distance between an approximate and exact solution. For example a network with 12 channels, each channel with 9 possible capacities required an execution time by an exact algorithm of a few hours. Presented approximate algorithm for the same network gave an acceptable solution within 50 milliseconds.

Let $D^{\max}$ be the maximal building cost of the network and let $D^{\min}$ be the minimal building cost of the network; the problem (5-7) has no solution for $D < D^{\min}$.

T^{min} is average delay per packet obtained as solution of multicommodity flow problem for representation $X_1 = \{x_1^1, x_2^2, ..., x_1^b\}$. In other words T^{min} is the lowest possible value of average delay per packet at given requirement matrix. We introduce the normalized budget $\overline{B}$ and normalized coefficient $\overline{\alpha}$, which are used to compare the results received for different network topology, external traffic requirements and discrete cost-capacity function.

$$\overline{B} = \frac{B - D^{min}}{D^{max} - D^{min}} 100\% \quad , \quad \overline{\alpha} = \alpha \frac{T^{min}}{D^{max}} 100\% \tag{12}$$

Let $\Delta_Q = \frac{Q_{app} - Q_{opt}}{Q_{opt}} 100\%$ be the distance between approximate solution and exact solution, where: Q_{app} is value of combined cost criterion received as approximate solution of optimization problem (5-7). Q_{opt} is the value of combined cost criterion received as exact solution of optimization problem (5-7).

Let $\Phi_{(\overline{\alpha},\overline{B})} = \sum_{i=1}^{P} \frac{\Delta_Q^i}{i}$ be the arithmetic mean of the distance between an approximate and exact solution calculated for all considered networks and for normalized coefficient $\overline{\alpha}$ and normalized budget $\overline{B}$. P is the number of considered network topologies. The dependence of Φ on normalized budget $\overline{B}$ has been examined. In Table 1 values of Φ for different values of $\overline{B}$ and $\overline{\alpha}$ are presented. First we notice that approximate algorithm works very well as it is very close to optimal solutions for normalized budget greater than 40% and it is almost independent from value of α.

Table 1. Distances between exact and approximate solutions

	Normalized Budget					
Normalized $\overline{\alpha}$	10%	20%	30%	40%	50%	<60,100>
1%	6,8%	2,3%	1,5%	1,5%	1,3%	For normalize budget greater than 50 % value of Φ is lower than 1,6 %
2%	21,2%	4,3%	1,4%	1,3%	1,0%	
3%	36,3%	9,3%	1,6%	0,9%	1,0%	
10%	59,2%	18,7%	15,2%	1,6%	0,4%	
20%	104,6%	23,7%	22,0%	2,9%	0,8%	
40%	114,9%	30,2%	27,7%	3,5%	1,6%	
50%	124,7%	31,4%	29,2%	3,5%	1,7%	
80%	130,0%	32,0%	31,0%	4,0%	1,7%	

The worst results we obtained for normalized budged lower and equal to 30 % especially for $\overline{\alpha}$ greater than 10%. We notice that with increasing $\overline{\alpha}$ value of Φ increases as well. On Figure 1 the dependence of Φ on $\overline{B}$ and $\overline{\alpha}$ is presented.

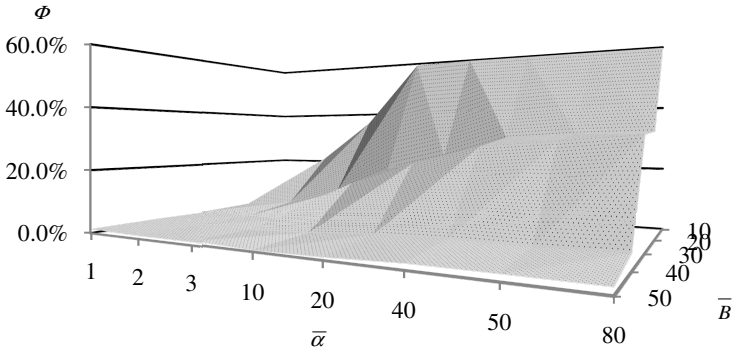


Fig. 1. The dependence of Φ on $\overline{B}$ and $\overline{\alpha}$

We performed many experiments with different network topologies, different set of possible capacities and various values of workload. A sample network topology named PL-02 is presented on Figure 2.

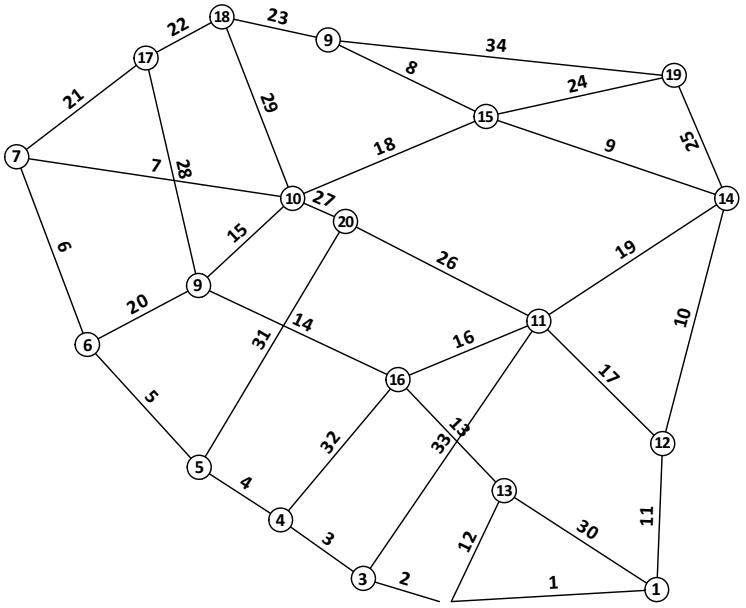


Fig. 2. PL-02 network topology

Presented network consists of 20 nodes and 34 channels. Nodes are located in 20 cities in Poland. For each channel we have 9 alternative capacities due to the line rates available to the customer by an ISP. The execution time of approximate algorithm is presented in Table 2. We have no exact solution for this network because for this scale of problem amount of possible solution reaches a number of 2.8^{32} which makes an exact algorithm infeasible because of unacceptable execution time.

Table 2. Executig time of approximate algorithm for network PL-02 in seconds

	Normalized Budget									
$\overline{\alpha}$	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1%	0,218	0,202	0,202	0,202	0,202	0,202	0,202	0,202	0,202	0,218
3%	0,171	0,141	0,140	0,140	0,156	0,140	0,156	0,140	0,140	0,156
5%	0,187	0,171	0,140	0,140	0,140	0,124	0,124	0,124	0,124	0,124
10%	0,156	0,140	0,109	0,062	0,062	0,078	0,078	0,078	0,078	0,078
20%	0,140	0,124	0,124	0,078	0,062	0,046	0,046	0,046	0,046	0,046
40%	0,156	0,140	0,109	0,062	0,062	0,062	0,046	0,031	0,046	0,046
60%	0,156	0,140	0,109	0,078	0,046	0,031	0,031	0,031	0,046	0,046
80%	0,156	0,140	0,093	0,078	0,062	0,046	0,046	0,046	0,046	0,031

We notice that an approximate algorithm is especially effective regarding the running time for normalized $\overline{\alpha}$ greater than 5% and for normalized budget greater than 30%. For this range executing time of algorithm is lower than 100ms. Otherwise for rest of the cases executing time is quite acceptable from practical point of view.

6 Conclusion

Capacity and Flow Assignment Problem is of great importance in contemporary computer networks. Recently emerging technologies and concepts like Software Defined Networks with its centralization and distinction between data and control plane provide a single place to provision bandwidth across the network. The complexity of an exact algorithm to CFA makes it infeasible to use, for that reason a heuristic algorithm was proposed in this paper. The experiments involved using an off-the-shelf desktop computer and a real-life network topology. It was shown, that if the normalized budget is scoped wide enough to search for the solution effectively, one is found within 0.2 of a second. Such running time makes it a viable solution for an SDN controller to provision bandwidth across a network.

References

1. Agarwal, S., Kodialam, M., Lakshman, T.V.: Traffic engineering in software defined networks. In: Proceedings of the IEEE INFOCOM, pp. 2211–2219. IEEE (2013)
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT Press (2001)
3. Gola, M., Czubak, A.: Designing Frame Relay WAN Networks with Trade-off Between Link Cost and Performance. In: Gruca, A., Czachórski, T., Kozielski, S. (eds.) Man-Machine Interactions 3. AISC, vol. 242, pp. 559–566. Springer, Heidelberg (2014)
4. Ferreira, R.P.M., Luna, H.P.L., Mahey, P., Souza, M.C.D.: Global optimization of capacity expansion and flow assignment in multicommodity networks. Pesquisa Operacional 33(2), 217–234 (2013)
5. Fratta, L., Gerla, M., Kleinrock, L.: The flow deviation method: An approach to store-and-forward communication network design. Networks 3(2), 97–133 (1973)
6. Gavish, B., Neuman, I.: A system for routing and capacity assignment in computer communication networks. IEEE Transactions on Communications 37(4), 360–366 (1989)
7. Girgis, M., Mahmoud, T., El-Hameed, H.A., El-Saghier, Z.: Routing and capacity assignment problem in computer networks using genetic algorithm. Inf. Sci. Lett 2, 13–25 (2013)
8. Gladysz, J., Walkowiak, K.: Combinatorial optimization of capacity and flow assignment problem for unicast and anycast connections with linear and convex objective functions–exact and heuristic algorithms. Electronic Notes in Discrete Mathematics, pp. 1041–1048 (2010)
9. Gola, M.: An algorithms for capacity and flow assignment in wide area computer network with combined cost criterion. Ph.D. thesis, Wroclaw University of Technology (2000)
10. Goscien, R., Pozniak-Koszalka, I., Koszalka, L., Kasprzak, A.: A Top-Down Heuristic for TCFA Problem in WAN. In: The Ninth International Conference on Wireless and Mobile Communications, ICWMC 2013, pp. 89–94 (2013)
11. Koldehofe, B., Dürr, F., Tariq, M.A., Rothermel, K.: The power of software-defined networking: line-rate content-based routing using OpenFlow. In: Proceedings of the 7th Workshop on Middleware for Next Generation Internet Computing, p. 3. ACM (2013)
12. Kasprzak, A.: Topological Design of the Wide Area Networks. Wroclaw University of Technology Press, Wroclaw (2001)
13. Kim, H., Feamster, N.: Improving network management with software defined networking. IEEE Communications Magazine 51(2), 114–119 (2013)
14. Lin, G., Huang, C., Zhan, S., Lu, X., Lu, Y.: Ranking Based Selection Genetic Algorithm for Capacity Flow Assignments. In: Cai, Z., Tong, H., Kang, Z., Liu, Y. (eds.) ISICA 2010. CCIS, vol. 107, pp. 97–107. Springer, Heidelberg (2010)
15. Pioro, M., Medhi, D.: Routing, flow, and capacity design in communication and computer networks. Morgan Kaufmann (2004)
16. Shen, J., Xu, F., Zheng, P.: A tabu search algorithm for the routing and capacity assignment problem in computer networks. Computers & Operations Research 32(11), 2785–2800 (2005)
17. Walkowiak, K.: A flow deviation algorithm for joint optimization of unicast and anycast flows in connection-oriented networks. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part II. LNCS, vol. 5073, pp. 797–807. Springer, Heidelberg (2008)
18. Zhan, Y., Lu, J., Li, S.: A Hybrid GA-TS Algorithm for Optimizing Networked Manufacturing Resources Configuration. Appl. Math 7(5), 2045–2053 (2013)

Evolution of Software Quality Models in Context of the Standard ISO 25010

Oleksandr Gordieiev[1], Vyacheslav Kharchenko[2], Nataliia Fominykh[1], and Vladimir Sklyar[3]

[1] Sevastopol Institute of Banking of University of Banking of the National Bank of Ukraine, 6 Parkova street, Sevastopol, Ukraine
alex.gordeyev@gmail.com
[2] National Aerospace University «KhAI», 17 Chkalova street, Kharkiv, Ukraine
V.Kharchenko@khai.edu
[3] Research and Production Corporation Radiy, 29 Geroev Stalingradu , Kirovpgrad, Ukraine
v.sklyar@radiy.com

Abstract. Evolutionary analysis of software (SW) quality models (QM) over the past forty years, from one of the first software QM by McCall to the model presented in the standard ISO 25010 is performed. 9 models were chosen for the analysis and divided into sets of basic and corporate QMs according to the completeness, detailing and significance. The choice of basic models McCall (1977), IEEE 1219 (1993), ISO9126-1 (2001), ISO 25010 (2010) is grounded. QM structure is described by hierarchy whose elements are sets of characteristics (subcharacteristics) and relations of subordination between them. To assess the complexity and completeness of SW QM and to compare them with the latest ISO 25010 model special particular and general metrics are introduced. Analytic dependence of the growth of model complexity represented by a linear function is obtained. Analysis of some characteristics evolution (operational suitability, effectiveness, reliability, usability, safety, etc) is performed.

Keywords: evolutionary analysis, software quality models, complexity metrics, ISO 25010.

1 Introduction

1.1 Motivation

Beginning of active development and use of software as an integral part of computers can be considered the middle of the last century. The term Software Engineering appeared in 1968 at NATO Software Engineering Conference [1] where corresponding concept was formed. During the period of software engineering development as an independent direction in Engineering practice, and then forming it as a systematic science, one of the key issues was software (SW) quality.

Software quality is a degree to which a software product satisfies stated and implied needs when used under specified conditions [2]. SW quality model is [3] usually defined as a set of characteristics and relationships between them which actually

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_21, © Springer International Publishing Switzerland 2014

provides the basis for specifying the requirements of quality and evaluating quality. A lot of software quality models (SWQM) have been introduced for the last decades [4]. Quality models structure is described by hierarchy whose elements are sets of characteristics (CHs) (subcharacteristics (SubCHs)) and relations of subordination between them. Characteristics (subcharacteristics) included into the models as usual are the basis of software projects requirements.

The motive of writing this paper is release of ISO25000 series as a new generation of International Organization for Standardization requirements document connected with software standardization and its quality evaluation. It was not just a very important step to improve SW QM, but a significant event in models evolution that reflected changes in software engineering as well.

1.2 Analysis Methods of SW Quality Models

A lot of SW quality models have been introduced for almost half century history of Software Engineering, but only a part of them became widely known and is used requirements elicitation. Preliminary analysis of the publications describing software quality models and techniques of SW QM analysis [2-14], allowed determining the most significant of them. Information about these models indicating the bibliographic data and two important parameters (quality hierarchy number and characteristics/subcharacteristics number levelization) is systemized in table 1.

Table 1. The most well-known SW Quality Models

№	SW QM name	Publication year	Model levels number	CHs / SubCHs number	Author	Reference/ Resource
1	McCall	1977	2	11/35	John McCall	[5]
2	Boehm	1978	3	3/8/18	Boehm	[6]
3	Carlo Ghezzi	1991	1	8	Carlo Ghezzi	[7]
4	FURPS	1992	2	5/25	Grady R. & Hewlett Packard	[8]
5	IEEE	1993	2	6/19	IEEE	[9]
6	Dromey	1995	2	4/13	Dromey	[10]
7	ISO 9126-1	2001	2	6/19	ISO	[13]
8	QMOOD	2002	1	6	Bansiya	[12]
9	ISO 25010	2010	2	8/31	ISO	[2]

SW quality models diversity causes a lot of works aimed at their comparative analysis. These researches intensified in the 90 years before ISO 9126 publication and in first years of its practical use.

Review of works related to SW QM analysis techniques [3, 12-14] shows that modern ISO 25010 standard, where a new SW quality model was adopted by the international institution, is not taken into account. Besides:

– The works are usually reduced to an analysis of characteristics and subcharacteristics are not considered enough or are not a subject of the analysis at all;

– Formal techniques of SW QM description are based on table representation of models, but their description in sets and relations between them is not analyzed;

– Procedure of SW QM analysis is not formalized.

Considering the importance of quality models base, it requires further researches in this field.

1.3 Problem Statement

Every new model appearing is a reflection of the quality of the constantly changing (rising, expanding and specifying) software requirements. It is caused by:

– Dynamic development of software products and technologies;
– Rising impact of software tools on functional capabilities, performance, reliability, security and other characteristics of the computer systems, and the technical systems in general;
– Development of the regulatory framework at the international, national and corporate levels.

Variety of software quality models and frequency of their appearance encouraged the authors to analyze the evolution of the well-known software quality models in the context of ISO 25010.

Regarding this, the goal of the paper is, firstly, to develop a formalized description of the SW QM and metrics to present them in a compact form, and, secondly, to perform a comparative analysis of software quality models and to study their changes for more than 40 years of evolution.

2 Systematic SW QM Description

2.1 Description Principles

The primary task while analyzing the evolution of software quality models is the problem of transition from their verbal and structural-table representation to the formal description in terms of the algebra of sets and relations. To describe the models, sets of software quality model elements (SSQME - Set of Software Quality Model Elements) and relationships of model elements (SRSQME - Set of Relationships Software Quality Model Elements) were defined. Set of software quality model elements may be presented as the following:

$$SSQME_i = \left\{EM_i^j, EM_i^{j+1}, ..., EM_i^n\right\}, \quad (1)$$

where i is index of model, j is index of model element EM_i^j.

To describe the relationship between the characteristics of different levels predicate (relation) R (“parent-child”) is used:

$$EM_i^j R EM_i^k, \quad (2)$$

where EM_i^j - parent, EM_i^k - child.

Thus, every software quality model (SQM) is described by two sets (the set of model elements and the set of model elements relations - R) and the table of the model semantic content is needed to determine SSQME sets elements accordance and their names. For example, let us describe ISO 25010 software quality model using this approach:

$$SSQME_9 = \left\{ \begin{array}{l} EM_9^1, EM_9^2, EM_9^3, EM_9^4, EM_9^5, EM_9^6, EM_9^7, EM_9^8, EM_9^9, EM_9^{10}, \\ EM_9^{11}, EM_9^{12}, EM_9^{13}, EM_9^{14}, EM_9^{15}, EM_9^{16}, EM_9^{17}, EM_9^{18}, EM_9^{19}, EM_9^{20}, \\ EM_9^{21}, EM_9^{22}, EM_9^{23}, EM_9^{24}, EM_9^{25}, EM_9^{26}, EM_9^{27}, EM_9^{28}, EM_9^{29}, EM_9^{30}, \\ EM_9^{31}, EM_9^{32}, EM_9^{33}, EM_9^{34}, EM_9^{35}, EM_9^{36}, EM_9^{37}, EM_9^{38}, EM_9^{39}, EM_9^{40} \end{array} \right\}$$

$$SRSQME_9 = \left\{ \begin{array}{l} EM_9^1 REM_9^2, EM_9^1 REM_9^6, EM_9^1 REM_9^{10}, EM_9^1 REM_9^{13}, EM_9^1 REM_9^{20}, \\ EM_9^1 REM_9^{25}, EM_9^1 REM_9^{31}, EM_9^1 REM_9^{35}, EM_9^2 REM_9^3, EM_9^2 REM_9^4, \\ EM_9^2 REM_9^5, EM_9^6 REM_9^7, EM_9^6 REM_9^8, EM_9^6 REM_9^9, EM_9^{10} REM_9^{11}, \\ EM_9^{10} REM_9^{12}, EM_9^{13} REM_9^{14}, EM_9^{13} REM_9^{15}, EM_9^{13} REM_9^{16}, EM_9^{13} REM_9^{17}, \\ EM_9^{13} REM_9^{18}, EM_9^{13} REM_9^{19}, EM_9^{20} REM_9^{21}, EM_9^{20} REM_9^{22}, EM_9^{20} REM_9^{23}, \\ EM_9^{20} REM_9^{24}, EM_9^{25} REM_9^{26}, EM_9^{25} REM_9^{27}, EM_9^{25} REM_9^{28}, EM_9^{25} REM_9^{29}, \\ EM_9^{25} REM_9^{30}, EM_9^{31} REM_9^{32}, EM_9^{31} REM_9^{33}, EM_9^{31} REM_9^{34}, EM_9^{35} REM_9^{36}, \\ EM_9^{35} REM_9^{37}, EM_9^{35} REM_9^{38}, EM_9^{35} REM_9^{39}, EM_9^{35} REM_9^{40} \end{array} \right\}$$

Its semantic content is presented by table 2.

Table 2. ISO 25010 Model Semantic Content Table

SSQME element	Model element	SSQME element	Model element	SSQME element	Model element
EM_9^1	System/Software Product Quality	EM_9^{15}	4.2 Learnability	EM_9^{28}	6.3 Non-repudiation
EM_9^2	1 Functional Suitability	EM_9^{16}	4.3 Operability	EM_9^{29}	6.4 Accountability
EM_9^3	1.1 Functional Completeness	EM_9^{17}	4.4 User Error Protection	EM_9^{30}	6.5 Authenticity
EM_9^4	1.2 Functional Correctness	EM_9^{18}	4.5 User Interface Aesthetics	EM_9^{31}	7 Portability
EM_9^5	1.3 Functional Appropriateness	EM_9^{19}	4.6 Accessibility	EM_9^{32}	7.1 Adaptability
EM_9^6	2 Performance Efficiency	EM_9^{20}	5 Reliability	EM_9^{33}	7.2 Installability
EM_9^7	2.1 Time Behaviour	EM_9^{21}	5.1 Maturity	EM_9^{34}	7.3 Replaceability
EM_9^8	2.2 Resource Behaviour	EM_9^{22}	5.2 Availability	EM_9^{35}	8. Maintainability
EM_9^9	2.3 Capacity	EM_9^{23}	5.3 Fault Tolerance	EM_9^{36}	8.1 Modifiability
EM_9^{10}	3 Compatibility	EM_9^{24}	5.4 Recoverability	EM_9^{37}	8.2 Testability
EM_9^{11}	3.1 Co-existence	EM_9^{25}	6 Security	EM_9^{38}	8.3 Modularity
EM_9^{12}	3.2 Interoperability	EM_9^{26}	6.1Confidentiality	EM_9^{39}	8.4 Reusability
EM_9^{13}	4 Usability	EM_9^{27}	6.2 Integrity	EM_9^{40}	8.5 Analyzability
EM_9^{14}	4.1 Appropriateness recognisability				

For the whole set of software quality models SSQM = {SQM_i} a common set of model elements using a simple sets union operation can be formed. However, this requires further analysis of the semantic content of every characteristic and subcharacteristic which is not being carried out in this article. For this reason, additional elements of indexing are used within the models. It should be noted that model level number for the formal description of software quality models isn't directly considered.

3 Metrics

To briefly characterize the proposed analysis technique, let us introduce some initial terms:

– Conceptual model is a model which a model under study is compared with;
– Model under study is a model which is compared with a conceptual model;
– Characteristic under study is a conceptual model characteristic which is compared with model under study characteristics.

The technique is based on comparing a model under study with the conceptual model, i.e. every SW Quality Model is compared with the conceptual model. So, the analysis is equivalent to semantic comparing characteristics and subcharacteristics of a model under study and the conceptual model with regard to their structures. Selecting a reference model, is usually performed by an expert who has relevant experience and qualifications.

At the following stage comparison of models among themselves should be performed. The simplest and most obvious metrics are offered. Relationship and subordination of these metrics is presented in Fig. 1.The main objective of such metrics is comparison of models with reference model bottom up, i.e. at the level of subcharacteristics (SMM, CSCM and CMM metrics), further characteristics (CMCM metric) and models as a whole (CSQMCM metric).

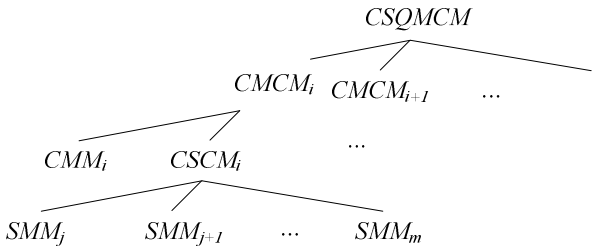


Fig. 1. Metrics relation and subordination

Features of the proposed metrics are the following:

– Subcharacteristics matching metric (SMMj). Every subcharacteristic match value is identified according to the following formula SMMj = 0,5 / number of reference (conceptual) model elements subcharacteristics of the characteristic under study;

– Cumulative subcharacteristics comparison metric (CSCM) is evaluated as a sum of SMM:

$$CSCM_i = \sum_{j=1}^{k} SMM_j ; \quad (3)$$

– Characteristics matching metric (CMM) takes the value of 0.5 in case of matching or 0 if the characteristics are different;

– Cumulative matching characteristics metric (CMCM) is calculated as a sum of CMM metric and $\sum_{j=1}^{k} CSCM_j$:

$$CMCM_i = CMM_i + \sum_{j=1}^{k} CSCM_j ; \quad (4)$$

– Cumulative software quality models comparison metric (CSQMCM) is calculated according to the formula:

$$CSQMCM_i = \sum_{i=1}^{n} CMCM_i . \quad (5)$$

4 Software Quality Models Analysis Results

Let us conduct SW QM analysis and first of all, define the reference (contextual) model. SW Quality Model ISO 25010 will be considered as uppermost and etalon regarding to all other models. It is the newest introduced model and takes into account main modern software peculiarities in point of view quality evaluation. This model is described in an international standard of top level. The reference model characteristics set consists of functional suitability, performance efficiency, compatibility, usability, reliability, security, portability, maintainability.

Results of semantic comparison of software quality models characteristics and subcharacteristics for different levels are presented in table 3.

Starting abscissa point (SW QM appearance years) is 1970. To represent information in a compact form characteristics and subcharacteristics are presented by numbers in table 3 according with table 2 for SW QM ISO 25010.

The results of quality models characteristics analysis are the following. Eevery 10 years a new quality model appears. The considered SW QMs can be divided into the following groups:

– the first group consists of the fundamental basic SW QM. These models are the result of authoritative international team work, such as ISO and IEEE. Therefore, such SW QM as IEEE, ISO 9126-1, ISO 25010 are regarded as fundamental;

– the second group consists of corporative SW QM. These models, as usual, are significant (signature models) and according to quality level (nomenclature, characteristics and their relations) and significantly inferior in quality to the basic models. McCall, Ghezzi, FURPS, Dromey, QMOOD are considered to be in this group.

Results of the models comparison using CSQMCM metrics (table 2) are presented by diagram (Fig. 2). Values of CSQMCM metrics for each model are presented in table 3.

Table 3. Software Quality Models Analysis Results

Conceptual model (ISO 25010)		**ISO 9126 model**				**QMOOD model**			
CHs	Sub-CHs	CHs	Sub-CHs	CMM	SMM	CHs	Sub-CHs	CMM	SMM
1		1	-	0,5	0	4	-	0,5	0
	1.1	-	-	0	0	-	-	0	0
	1.2	-	1.2	0	0,167	-	-	0	0
	1.3	-	-	0	0	-	-	0	0
				CMCM=0,667				**CMCM=0,5**	
2		4		0,5	0	6	-	0,5	0
	2.1	-	4.1	0	0,167	-	-	0	0
	2.2	-	4.2	0	0,167	-	-	0	0
	2.3	-	-	0	0	-	-	0	0
				CMCM=0,834				**CMCM=0,5**	
3		-	-	0	0	-	-	0	0
	3.1	-	6.3	0	0,25	-	-	0	
	3.2	-	1.3	0	0,25	-	-	0	0
				CMCM=0,5				**CMCM=0**	
4		3	-	0,5	0	-	-	0	0
	4.1	-	-	0	0	-	-	0	0
	4.2	-	3.2	0	0,083	-	-	0	0
	4.3	-	3.3	0	0,083	-	-	0	0
	4.4	-	-	0	0	-	-	0	0
	4.5	-	3.4	0	0,083	-	-	0	0
	4.6	-	-	0	0	-	-	0	0
				CMCM=0,749				**CMCM=0**	
5		2	-	0,5	0	-	-	0	0
	5.1	-	2.1	0	0,125	-	-	0	0
	5.2	-	-	0	0	-	-	0	0
	5.3	-	2.2	0	0,125	-	-	0	0
	5.4	-	2.3	0	0,125	-	-	0	0
				CMCM=0,875				**CMCM=0**	
6		-	1.4	0	0,5	-	-	0	0
	6.1	-	-	0	0	-	-	0	0
	6.2	-	-	0	0	-	-	0	0
	6.3	-	-	0	0	-	-	0	0
	6.4	-	-	0	0	-	-	0	0
	6.5	-	-	0	0	-	-	0	0
				CMCM=0,5				**CMCM=0**	
7		6	-	0,5	0	-	-	0	0
	7.1	-	6.1	0	0,166	-	-	0	0
	7.2	-	6.2	0	0,166	-	-	0	0
	7.3	-	6.4	0	0,166	-	-	0	0
				CMCM=0,998				**CMCM=0**	
8		5	-	0,5	0	-	-	0	0
	8.1	-	5.2	0	0,1	-	-	0	0
	8.2	-	5.4	0	0,1	-	-	0	0
	8.3	-	-	0	0	-	-	0	0
	8.4	-	-	0	0	1	-	0,1	0
	8.5	-	5.1	0	0,1	-	-	0	0
				CMCM=0,8				**CMCM=0,1**	
		CSQMCM=5,9				**CSQMCM=1,1**			

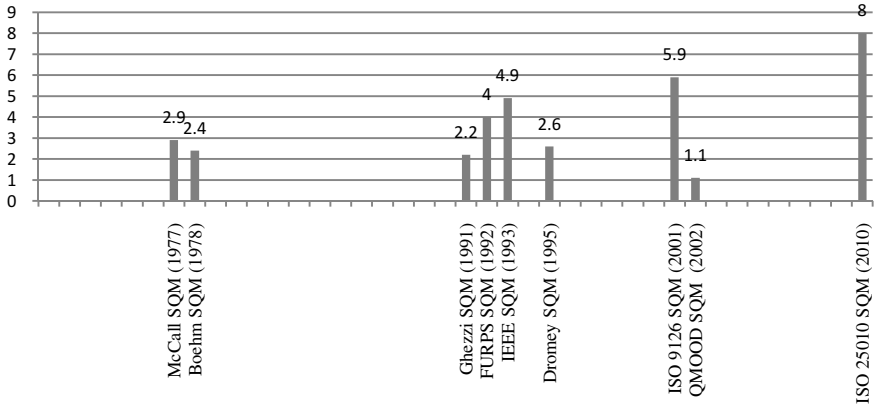


Fig. 2. Results of SW QM comparison in terms of CSQMCM

Further, we consider the models for which the values of the CSQMCM metric were higher than the previous once according to the chronology of their emergence (Fig. 3). These models such as McCall's, IEEE, ISO 9126, ISO 25010 are landmark.

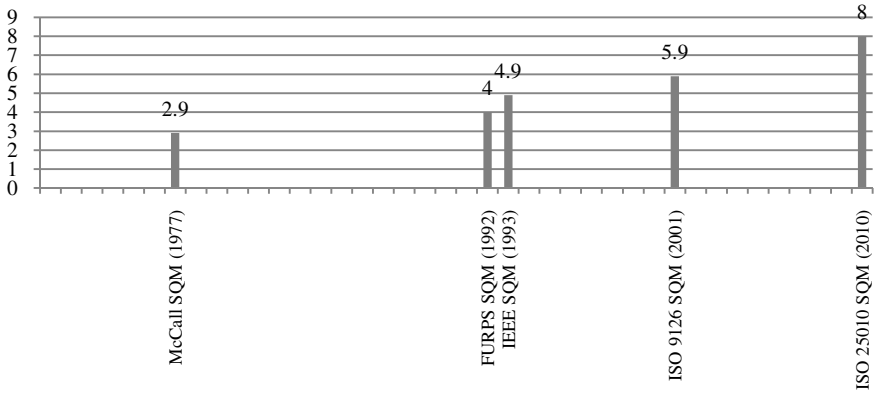


Fig. 3. Results of basic SW QM comparison in terms of CSQMCM

Let us define the functional connection between SW QM appearance year (X axis) and CSQMCM metric value (Y axis) and represent it analytically by regressive liner relationship:

$$y = ax+b, \tag{6}$$

where y is function, x - variable, a and b - regression coefficients.

The values of a and b parameters can be calculated using Least Square Method. As a result we have a = 0,153, b = 1,363 and liner dependence:

$$y = 0{,}153x+1{,}363. \tag{7}$$

To analyze the changes of SW QM on the level of characteristics let's match CMCM metric value for SW QM (table 4) and represent them in table 4.

Table 4. CMCM Metric Value for basic SW QM

SW QM / Characteristics	McCall (1977), CMCM	FURPS (1992), CMCM	IEEE (1993), CMCM	ISO 9126-1 (2001), CMCM	ISO 25010 (2010), CMCM
Functional Suitability	0	0,5	0,83	0,67	1
Performance Efficiency	0,5	0,84	0,67	0,83	1
Compatibility	0	0,5	0,75	0,5	1
Usability	0,666	0	0,58	0,75	1
Reliability	0,625	0,75	0,75	0,88	1
Security	0	0,5	0,5	0,5	1
Portability	0,5	0,334	0,67	0,99	1
Maintainability	0,6	0,6	0,1	0,8	1

5 Conclusions

As a result, structural and semantic analysis technique, based on their set-theoretic description and special metrics, is proposed. This technique allows obtaining metrics values, needed for quantitative SW QM comparison that made possible evolutionary changes of SW QM.

Considering completeness and integrity metrics two SW QM groups were defined: basic and corporative. The proposed technique can be used as a basis to adapt the existed SW QMs for software companies.

Obtained liner dependence between CSQMCM metrics and SW QM appearance year describes some pattern of liner growth of SW QM complexity.

As for further development of SW QM, it is possible to make the following predictions:

– CSQMCM metric for every next model must be higher than for the previous one, i.e. characteristics nomenclature will be wider;

– SW QM characteristics structure will become more and more complicated due to their further development and specification on the subcharacteristics level, especially for certain characteristics such as reliability, security, usability and others.

It is planned for further work:

– To conduct more detailed semantic analysis of all the characteristics (subcharacteristics) in term of their unambiguous interpretation and characteristics (subcharacteristics) duplication within the same SW QM;

– To investigate metrics evolution to assess SW quality characteristics, evaluation methods and techniques;

– To perform evolutionary analysis of SW QMs separately for dependability, security, usability, etc. for different domains.

References

1. Software engineering. Report on a conference sponsored by the NATO SCIENCE COMMITTEE, Garmisch, Germany (October 7-11, 1968), http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF
2. ISO/IEC 25010: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, ISO/IEC JTC1/SC7/WG6 (2011)
3. Dubey, S.K., Ghosh, S., Rana, A.: Comparison of Software Quality Models: An Analytical Approach. International Journal of Emerging Technology and Advanced Engineering 2(2), 111–119 (2012)
4. Wagner, S.: Software Product Quality Control. Springer, Heidelberg (2013)
5. McCall, J.A., Richards, P.K., Walters, G.F.: Factors in Software Quality. Nat'l Tech. Information Service 1, 2 and 3 (1977)
6. Boehm, B.W., Brown, J.R., Kaspar, H., Lipow, M., McLeod, G., Merritt, M.: Characteristics of Software Quality. North Holland, Amsterdam (1978)
7. Ghezzi, C., Jazayeri, M., Mandrioli, D.: Fundamentals of Software Engineering. Prentice Hall, New Jersey (1991)
8. Grady, R.B.: Practical software metrics for project management and process improvement. Prentice Hall, New Jersey (1992)
9. IEEE, std 1219: Standard for Software Maintenance. IEEE Computer Society Press, USA (1993)
10. Dromey, G.R.: A model for software product quality. Transactions of Software Engineering 21(2), 146–162 (1995)
11. Hyatt, L.E., Rosenberg, L.H.: A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. In: ESA 1996 Product Assurance Symposium and Software Product Assurance Workshop, ESTEC, Noordwijk, The Netherlands, European Space Agency, pp. 209–212 (1996)
12. Bansiya, J., Davis, C.: Hierarchical Model for Object-Oriented Quality Assessment. IEEE Transactions on Software Engineering 28, 4–17 (2002)
13. ISO/IEC 9126-1 Software engineering – Product quality – Part 1: Quality model (2001)
14. Gordeyev, A., Kharchenko, V., Andrashov, A., Sklyar, V., Konorev, B., Boyarchuk, A.: Case-based Software Reliability Assessment by Fault Injection Unified Procedures. In: Proc. of Software Engineering in East and South Europe (SEESE 2008), Leipzig, Germany, May 13, pp. 1–8. ACM, New York (2008)

Model Checking of UML Activity Diagrams in Logic Controllers Design

Iwona Grobelna[1], Michał Grobelny[2], and Marian Adamski[1]

[1] University of Zielona Góra, Institute of Computer Engineering and Electronics, Zielona Góra, ul. Szafrana 2, Poland
{i.grobelna,m.adamski}@iie.uz.zgora.pl
[2] University of Zielona Góra, Department of Media and Information Technologies, Zielona Góra, al. Wojska Polskiego 69, Poland
m.grobelny@kmti.uz.zgora.pl

Abstract. The article presents a novel approach to model checking of UML activity diagrams (in version 2.x) for logic controller specification. A novel idea to design embedded systems by means of activity diagrams is introduced, using the previously proposed rule-based logical model suitable both for formal verification and logic synthesis. As the result implemented solution is consistent with the verified specification delivered in form of an user-friendly UML activity diagram. The idea is presented on a simple control process of two vehicles movement. Model checking technique is used to verify system model against behavioral properties expressed in temporal logic. In case of detected errors appropriate counterexamples are generated.

Keywords: UML activity diagram, rule-based logical model, formal verification, temporal logic, synthesis.

1 Introduction

The Unified Modelling Language (UML) [1] is a user-friendly, established technique for modelling software. Commonly, it is also used in logic controller design process [2-4]. Especially valuable are here UML state machines and UML activity diagrams. The article focuses particularly on activity diagrams as a behavioral specification of designed logic controller.

Former researches have shown that is possible to specify the behavior of embedded systems using UML activity diagrams [2,3]. Unfortunately, UML diagrams are still not well supported by formal techniques for analysis and verification. On the other hand, control interpreted Petri nets [5], as a mathematical apparatus, have a wide-range support of techniques and mechanisms which can improve the quality of specification. It is possible to transform an UML activity diagram into a control Petri net [3] and then use the benefits of the second specification technique, using the previously proposed rule-based logical model.

The rule-based logic model [6,7] is suitable both for formal verification [8] (using model checking technique [9,10]) and for synthesis. It includes rules describing the

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_22,

behavior of logic controller, and has been primarily based on a control interpreted Petri net. Properties of designed system to be verified are expressed with temporal logic formulas [11]. Model checking is performed using the *NuSMV* model checker [12]. The rule-based logical model is a basis for VHDL code, which can be easily synthesized and implemented in reconfigurable structures of FPGA-type [13,14].

The novel idea is to formally verify UML activity diagrams describing logic controller using the rule-based logical model. The logical model formalizes the UML diagram. It is then possible to verify it using model checking technique, and then to synthesize it. A big advantage of proposed solution is that logic controller can be easily specified in user-friendly form and at the same time the so-prepared specification can be formally verified and synthesized. So, after formal verification the quality of final product rises significantly and the implemented device is consistent with its primary specification.

There have already been some approaches to formalize UML activity diagrams [15-17]. In [15] authors also tried to analyze activity diagrams using the *NuSMV* model checker. However, much more code corresponding to particular elements is generated and additionally, activity diagrams are translated directly into the *NuSMV* model checker input format and so the possibility to use them for synthesis is lost. In [16] authors aim to define an algebraic presentation of the semantic of UML based on institution theory. In [17] a formal execution semantics for UML activity diagrams (in an older version) appropriate for workflow modelling is proposed. However, none of the so-far approaches formalized UML activity diagrams in domain of logic controllers design using temporal logic and a rule-based logical model suitable both for verification and for synthesis.

The remainder of the paper is structured as follows. Section 2 introduces a novel idea to present an UML activity diagram as a rule-based logical model suitable both for formal verification using model checking technique and for logical synthesis. Section 3 provides some information about model checking, properties definition and counterexamples. Section 4 focuses on logical synthesis of a specification primary expressed by means of UML activity diagrams. Finally, section 5 summarizes the paper and presents the results.

2 UML Activity Diagram as a Rule-Based Logical Model

UML activity diagram for discreet control process can be highly simplified defined as a seven-tuple $AD = \{A, T, G, F, S, E, Z\}$ where:

- A is a set of actions/activities;
- T is a set of transitions (i.e. fork and merge nodes);
- G is a set of guard conditions corresponding to transitions (input signals);
- F is a set of flow relation between the activities and transitions;
- S is a set containing an initial node;
- E is a set containing a final node;
- Z is a set of output signals.

A prepared activity diagram describing logic controller behavior can be written as a rule-based logical model (Fig. 1). Logical model [6,7] can be then formally verified using model checking technique. It can be automatically transformed into verifiable model (using the implemented *m2vs* tool). The *NuSMV* tool [12] checks therefore whether the defined system model satisfies list of requirements specified as temporal logic formulas. On the other hand, logical model can be also transformed into a synthesizable model in VHDL language and then simulated and synthesized. The most important is here the fact, that both verifiable and synthesizable models are consistent with each other. Hence, the received physical implementation is consistent with the primary, already verified, specification.

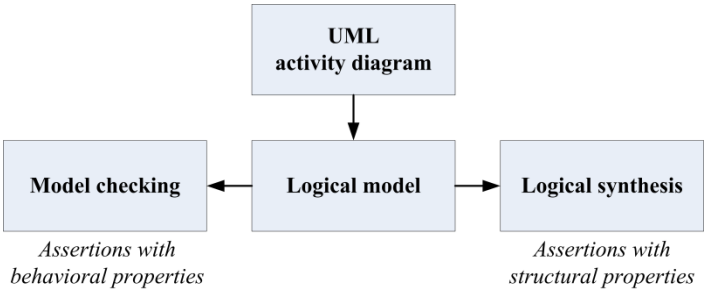


Fig. 1. Verification and synthesis of UML activity diagram with intermediate logical model

2.1 Rule-Based Logical Model

A rule-based logical model has been initially established for control interpreted Petri nets specifying logic controller behavior [6,7]. However, it is also well-suited for use on the basis on UML activity diagrams. It consists of several sections corresponding to particular elements of specification.

Logical model starts with defined variables (keyword **`VARIABLES`**), which are assigned some predefined initial values (keyword **`INITIALLY`**). The following elements of UML activity diagram are interpreted as model variables:

- Actions (*A*) together with initial (*S*) and final (*E*) node
  Each action is assigned a variable of Boolean type. It takes the *TRUE* value if the action is already executed. It is possible that multiple actions have the *TRUE* value assigned, as there may be several concurrent processes defined. Additionally, initial and final nodes are taken into account to establish the initial and final state of the system (respectively).
- Input signals (*G*)
  Each input signal of logic controller is assigned a variable of Boolean type. It takes the *TRUE* value if the signal is active and the *FALSE* value otherwise. It is possible that multiple input signals have the *TRUE* value assigned, as there may be several input signals active at the same time.

– Output signals (Z)
  Each output signal of logic controller is assigned a variable of Boolean type. It takes the *TRUE* value if the signal is active and the *FALSE* value otherwise. It is possible that multiple output signals have the *TRUE* value assigned, as there may be several output signals active at the same time.

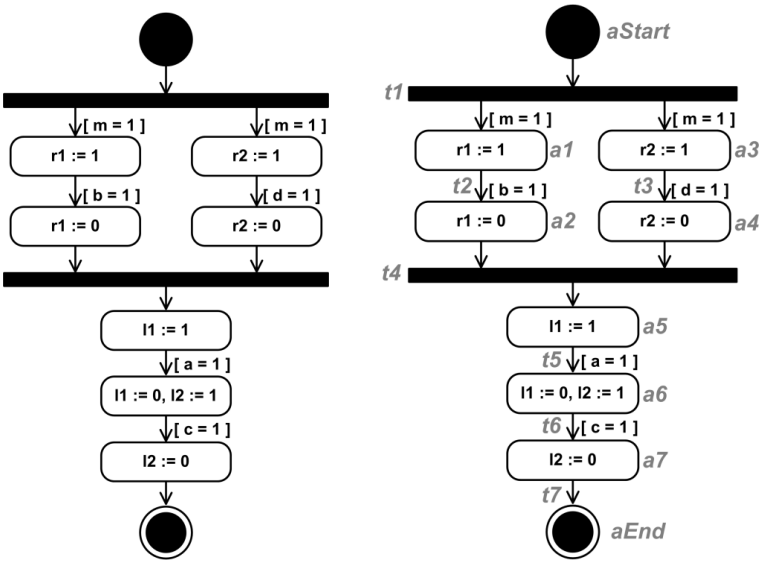
The set of transitions (T) and the set of flow relation (F) between the actions and transitions is used for rules definition (keyword **`TRANSITIONS`**).

The set of input signals G is mapped in section for inputs (keyword **`INPUTS`**), while the set of output signals – in section for outputs (keyword **`OUTPUTS`**).

2.2 Illustration

A sample UML activity diagram describing discreet process of two vehicles movement [3] is shown in Figure 2a. Initially, both vehicles are at starting points a and c. After pressing the m button, they begin to move to the right simultaneously until they reach ending points b and d (respectively). Then, the first vehicle moves left and returns to its starting point a. Afterwards, the second vehicle moves left and returns to its starting point c.

Each action $a \in A$ is labelled with an etiquette aX, where X stands for the number of action. Moreover, the initial node S is labelled as *aStart* and the final node E as *aEnd.* Each transition $t \in T$ is labelled with an etiquette tX, where X stands for the number of transition. The labelled activity diagram is presented in Figure 2b.



(a) (b)

Fig. 2. UML activity diagram for controlling two vehicles movement (a) and with labelling (b)

Basing on a labelled UML activity diagram, a rule-based logical model is built. An example for diagram from Figure 2 is listed in Figure 3. In initial values declaration the exclamation mark is used as a negation. Each transition starts with its etiquette to make the logical model more readable and simpler to interpret. The rules are written using temporal logic operator *X* for the next state of variables. Input signals changes, defined in logical model, are only used in model checking process (model description preparation). In the HDL (Hardware Description Language) file, input signals are not concerned as they are inputs to logic controller. They are supposed to change their value if a particular action (defined on the left side) is active. Hence, always two possible values of input signal are given: inactive signal / active signal.

```
VARIABLES
places:  aStart, a1, a2, a3, a4, a5, a6, a7, aEnd
inputs:  m, a, b, c, d
outputs: r1, r2, l1, l2
INITIALLY
aStart; !a1; !a2; !a3; !a4; !a5; !a6; !a7; !aEnd
!m; !a; !b; !c; !d
TRANSITIONS
t1: aStart & m -> X (!aStart & a1 & a3);
t2: a1 & b -> X (!a1 & a2);
t3: a3 & d -> X (!a3 & a4);
t4: a2 & a4 -> X (!a2 & !a4 & a5);
t5: a5 & a -> X (!a5 & a6);
t6: a6 & c -> X (!a6 & a7);
t3: a7 -> X (!a7 & aEnd);
INPUTS
aStart -> !m | m;
a1 -> !b | b;
a3 -> !d | d;
a5 -> !a | a;
a6 -> !c | c;
OUTPUTS
a1 -> r1;
a3 -> r2;
a5 -> l1;
a6 -> l2;
```

Fig. 3. Rule-based logical model of UML activity diagram

3 Model Checking of Activity Diagram

Logical model derived from UML activity diagram is transformed into format of the *NuSMV* model checker automatically using the *m2vs* tool according to the following rules (a snapshot from model description for considered example is shown in Fig. 4):

1. Each place is a variable of Boolean type.
   For each variable $p \in P$ a separate variable is defined `p : boolean`
2. Each input signal is a variable of Boolean type.
   For each variable $x \in X$ a separate variable is defined `x : boolean`
3. Each output signal is a variable of Boolean type.
   For each variable $y \in Y$ a separate variable is defined `y : boolean`
4. Defined variables take some initial values. Each variable takes any of two values *TRUE* or *FALSE*:
   `init(variable) := TRUE` or `init(variable) := FALSE`
5. Each place changes according to the rules defined in the transitions; conditions of changes between places occur in pairs (groups) – in the previous place(s) and in the next place(s)
6. Each input signal changes randomly, but can take the expected values connected with actions of UML activity diagram or change adequately to the situation;
7. Output signals changes are defined in correlation to actions of activity diagram.

```
next(aStart) := case
    aStart & m : FALSE;
    TRUE : aStart;
esac;
next(a1) := case
    a1 & b  : FALSE;
    aStart & m : TRUE;
    TRUE : a1;
esac;
next(a2) := case
    a2 & a4  : FALSE;
    a1 & b  : TRUE;
    TRUE : a2;
esac;
...
next(m) := case
    aStart : {FALSE, TRUE};
    TRUE : FALSE;
esac;
...
next(r1) := case
    a1 : TRUE;
    TRUE : FALSE;
esac;
...
```

Fig. 4. Model description in NuSMV of an UML activity diagram

Verifiable model description (in NuSMV format) starts with variables definition and their initial values. Then, each variable is assigned a value in the next state (shown in Fig. 4). So, in the example the *a1* variable (corresponding to *a1* activity in Fig. 2):

- becomes *TRUE* if both variables *aStart* and *m* are *TRUE*,
- becomes *FALSE* if both variables *a1* and *b* are *TRUE*,
- maintains its value otherwise.

Additionally, by input signals value changes (i.e. variable *m* in Fig. 4) one of possible values *FALSE* or *TRUE* is randomly chosen for the next state if a particular activity is already executed. Otherwise is maintains the *FALSE* value in order to eliminate the so-called state explosion problem.

To formally verify model description, requirements list is also needed. The list includes properties which are supposed to be fulfilled in designed embedded system. Requirements are defined using linear temporal logic (LTL) formulas, as i.e.:

$$\mathbf{G}\,(\mathrm{m} \rightarrow \mathbf{F}\ \mathrm{r1} \wedge \mathrm{r2}) \tag{1}$$

$$\mathbf{G}\,(\mathrm{b} \rightarrow \mathbf{F}\,\mathrm{l1}) \tag{2}$$

$$\mathbf{G}\,\neg(\mathrm{l1} \wedge \mathrm{r1}) \tag{3}$$

Formula (1) states that always when input signal *m* is active (button pressed), then finally both output signals *r1* and *r2* are active (both vehicles move to the right). Formula (2) states that always when input signal *b* is active (the first vehicle reached its ending point), then finally output signal *l1* is active (the first vehicle returns). Formula (3) states that is should never be the case that both output signals *l1* and *r1* are active at the same time (engines to move left and to move right for the first vehicle).

The most frequently verified properties regard liveness *(something good will eventually happen*), i.e. formulas (1) and (2), and safety (*something bad will never happen*), i.e. formula (3). In the verification process mostly behavioral requirements are taken into account which include input and output signals activity.

The *NuSMV* model checker compares model description with delivered requirements list and gives an answer whether properties are satisfied in designed system. If this is not the case, appropriate counterexamples are generated (as shown in Fig. 5 for formula (2), other properties are satisfied) which allow to find the error source in model description. Then either the specification or the requirement has been incorrectly formulated.

The sample counterexample (Fig. 5) shows that is possible that the first vehicle reaches its ending point *b*, but does not start to return (it does not move left). The sequence of states is presented which leads to the undesired situation. It can though happen that the second vehicle never reaches its ending point *d*, and so the vehicles cannot return to the starting positions.

```
-- specification  G (b ->  F l1)  is false
-- as demonstrated by the following execution sequence
Trace Description: LTL Counterexample
Trace Type: Counterexample
-> State: 1.1 <-                    -> State: 1.2 <-
  aStart = TRUE                       m = TRUE
  a1 = FALSE                        -> State: 1.3 <-
  a2 = FALSE                          aStart = FALSE
  a3 = FALSE                          a1 = TRUE
  a4 = FALSE                          a3 = TRUE
  a5 = FALSE                          m = FALSE
  a6 = FALSE                        -> State: 1.4 <-
  a7 = FALSE                          b = TRUE
  aEnd = FALSE                        r1 = TRUE
  m = FALSE                           r2 = TRUE
  a = FALSE                         -> State: 1.5 <-
  b = FALSE                           a1 = FALSE
  c = FALSE                           a2 = TRUE
  d = FALSE                           b = FALSE
  r1 = FALSE                        -- Loop starts here
  r2 = FALSE                        -> State: 1.6 <-
  l1 = FALSE                          r1 = FALSE
  l2 = FALSE                        -> State: 1.7 <-
```

Fig. 5. Generated counterexample

4 Synthesis on the Basis of Activity Diagram

Synthesis of logic controller specification be means of UML activity diagrams is possible using rule-based logical model as an intermediate format. The transformation of logical model into synthesizable model in VHDL language is done automatically using the *m2vs* tool. Prepared model can be then easily simulated (using i.e. *Active-HDL* environment) and synthesized (using i.e. *XILINX Plan Ahead* environment).

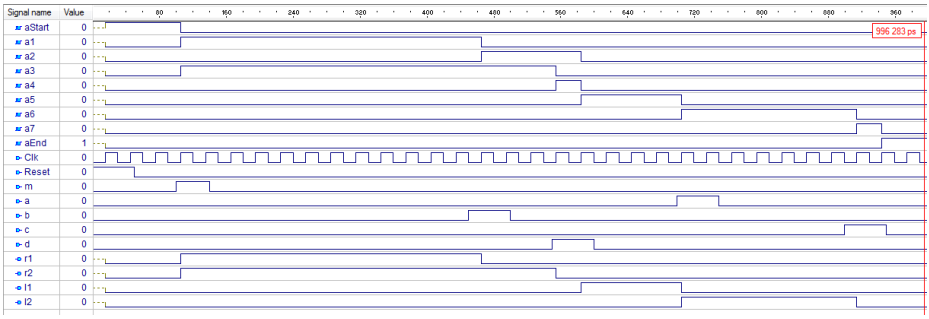


Fig. 6. Simulation results

Synthesis is performed in form of rapid prototyping [13,18] where optimization aspects are out of scope. Its main goal is to check whether designed system operates at all and some redundant hardware elements may be used.

The simulation results for considered example are presented in Figure 6. It is shown that designed logic controller behaves correctly according to the given specification.

5 Conclusions

The article presented a novel approach to formally verify logic controller specification by means of UML activity diagrams (in version 2.x). An original rule-based logical model is used to ensure the consistency between a verifiable and a synthesizable model. Formal verification is done using model checking technique and the *NuSMV* tool. Requirements to be then verified are expressed as temporal logic formulas. The transformation of a rule-based logical model into a verifiable and a synthesizable model is done automatically using the *m2vs* tool. The tool was developed to simplify the verification and synthesis process of logic controller specification and to ensure that the implemented solution is fully consistent with the previously verified specification. The rule-based logical model has to be prepared manually by a designer basing on a formal specification, which is here delivered by means of UML activity diagrams.

The results of the researches show that is possible to use the commonly-known and user-friendly UML language in logic controller design, focusing in particular on activity diagrams [2]. Using the proposed rule-based logical model [6,7], it is possible to formally verify the specification using model checking technique as well as to synthesize it in reconfigurable structures of FPGA-type. Finally, the implemented solution is consistent with the previously formally verified logic controller specification expressed by means of UML activity diagrams. Multiple examples of logic controllers specifications available in the literature have been successfully tested for behavioral properties, including i.e. models from [19,20]. Additionally, some studies have been also conducted in a local company of furniture industry, where some processes of furniture production are going to be automated using i.e. FPGA devices. Sample specifications of control processes have been prepared in form of UML activity diagrams which are easy understandable to non-technical partners. Then, the specifications have been formally verified using model checking technique and the proposed verification method with a rule-based logical model against some predefined behavioral properties.

Further research focuses on formal verification of UML activity diagrams with complex structures, including especially the hierarchy.

References

1. OMG Unified Modeling Language (OMG UML) Superstructure ver. 2.4.1. Object Management Group (2011)
2. Grobelny, M., Grobelna, I., Adamski, M.: Hardware behavioural modelling, verification and synthesis with UML 2.x activity diagrams. In: Proceedings of 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems (PDeS), Brno, pp. 109–114 (2012)

3. Grobelna, I., Grobelny, M., Adamski, M.: Petri Nets and activity diagrams in logic controller specification – transformation and verification. In: Proceedings of the 17th International Conference Mixed Design of Integrated Circuits and Systems, pp. 607–612 (2010)
4. Łabiak, G., Adamski, M., Doligalski, M., Tkacz, J., Bukowiec, A.: UML modelling in rigorous design methodology for discrete controllers. International Journal of Electronics and Telecommunications 58(1), 27–34 (2012)
5. David, R., Alla, H.: Discrete, Continuous, and Hybrid Petri Nets. Springer (2010)
6. Grobelna, I.: Formal verification of logic controller specification by means of model checking. Lecture Notes in Control and Computer Science, vol. 24. University of ZielonaGóra Press (2013)
7. Grobelna, I.: Formal verification of embedded logic controller specification with computer deduction in temporal logic. Przegląd Elektrotechniczny (12a), 40–43 (2011)
8. Kropf, T.: Introduction to Formal Hardware Verification. Springer (1999)
9. Clarke, E.M., Grumberg, O., Peled, D.A.: Model checking. The MIT Press (1999)
10. Emerson, E.A.: The Beginning of Model Checking: A Personal Perspective. In: Grumberg, O., Veith, H. (eds.) 25 Years of Model Checking: History, Achievements, Perspectives, pp. 27–45. Springer (2008)
11. Huth, M., Ryan, M.: Logic in Computer Science. Modelling and Reasoning about Systems. Cambridge University Press (2004)
12. Cavada, R., et al.: NuSMV 2.5 User Manual, `http://nusmv.fbk.eu/`
13. Ahrends, S.: Neue Ansätze für effizientes Rapid Prototyping von Embedded Systemen. In: Embedded Computing Conference (2008)
14. Wisniewski, R., Barkalov, A., Titarenko, L., Halang, W.A.: Design of microprogrammed controllers to be implemented in FPGAs. International Journal of Applied Mathematics and Computer Science 21(2), 401–412 (2011)
15. Lam, V.S.W.: A Formalism for Reasoning about UML Activity Diagrams. Nordic Journal of Computing 14(1), 43–64 (2007)
16. Achouri, A., Jemni Ben Ayed, L.: A Formal Semantic for UML 2.0 Activity Diagram based on Institution Theory. The International Journal of Soft Computing and Software Engineering (JSCSE) 3(3) (2013); Special Issue: The Proceedings of International Conference on Soft Computing and Software Engineering, USA
17. Eshuis, R., Wieringa, R.: A Formal Semantics for UML Activity Diagrams - Formalising Workflow Models. University of Twente, Centre for Telematics and Information Technology technical reports series (2001)
18. Andreu, D., Souquet, G., Gil, T.: Petri Net based rapid prototyping of digital complex system. In: IEEE Computer Society Annual Symposium on VLSI, pp. 405–410 (2008)
19. Adamski, M., Chodań, M.: Discreet control systems modelling using SFC nets. Wydawnictwo Politechniki Zielonogórskiej (2000) (in Polish)
20. Tkacz, J., Adamski, M.: Logic design of structured configurable controllers. In: IEEE 3rd International Conference on Networked Embedded Systems for Every Application (NESEA), pp. 1–6 (2012)

Impact of Selected Java Idioms on Source Code Maintainability – Empirical Study

Bogumiła Hnatkowska and Anna Jaszczak

Wrocław University of Technology, Institute of Informatics, Poland
bogumila.hnatkowska@pwr.wroc.pl, jaszczak.ania@gmail.com

Abstract. Source code maintainability is a desired software feature. It can be achieved in many different ways. For example, software engineers recommend the use of patterns as commonly known and proven quality solutions to existing problems. Patterns are usually defined at three basic granularity levels, and are classified as architectural, design, and programming patterns (idioms). The paper presents the results of an experiment conducted at Wroclaw University of Technology which aimed at checking the influence of Java idioms on source code maintainability. The obtained results confirmed that using idioms is a beneficial practice, especially in corrective maintenance.

Keywords: java, maintainability evaluation, programming patterns.

1 Introduction

The market of legacy systems is growing every year, and – in consequence – the number of programmers involved in system maintenance is increasing. Maintenance is the longest and the most expensive phase in software life cycle [6], so new techniques and methods, which aim at making the maintenance process more efficient, are developed all the time. One of the most popular maintenance techniques is refactorization that improves the internal source structure without changing the program behaviour. However, according to Pigoski [16], source code should be written with maintenance purpose in mind. In other words programmers should be encouraged to write "clean code" – code which is readable, simple, and direct [15].

Programming idioms belong to the elements of "clean code" and can be defined as a low level programming patterns [3]. "Idioms describe how to solve implementation-specific problems in programming language, such as memory management in C++" [3]. In some cases it is difficult to distinguish idioms from design patterns, so the following differences between them should be noted:

1. Idioms are used in the programming phase. Design patterns are applicable at the design phase.
2. Idioms have a smaller range than design patterns. To make the distinction clear we assume that one idiom affects at most one class.
3. Idioms are language-specific. Design patters are more general and most of them are language-independent.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_23,

"Idioms demonstrate competent use of programming language features", and can "support the teaching of programming language" [3]. However, the positive influence of programming idioms on program maintenance has not been documented by any significant study. The paper presents the results of an empirical study which aimed at answering the question if programming idioms have an impact on source code maintainability. Because programming idioms are specific to a given programming language, the selected Java idioms were the subject of investigation.

The paper is structured as follows. Section 2 provides an overview of Java idioms investigated in the study. Section 3 defines the maintainability notion, and places it in various quality models. In section 4 related works aiming at maintainability assessment are presented. The experiment itself is described in detail in Section 5. Collected data and their analysis is given is Section 6. The conclusions are derived and presented in the last Section 7.

2 Overview of Java Idioms

This chapter presents an overview of Java idioms found in different sources, e.g. [13], [4], [14], [8]. We limit the presentation only to the idioms that satisfy our criteria (an idiom affects at most one class). We selected more complex idioms (except better for-loop construct), which, in our opinion, you need to know to effectively use. Idioms are organized, following Bloch [4], into several categories. We tried to found at least 2 interesting idioms from each category.

1. Using interfaces: Interfaces for defining constants [13], Tag interface [13].
2. Creating and initializing objects: Static factory methods [4], Singleton [4], Test "whether" in constructor phase [13], Chain constructors [14].
3. Exception handling: Bouncer Pattern [13], Refactor Exception Handlers [13], Unhandled Exception [13], Convert Exceptions [13], Smart Exception [9], Tunnelling Exception [9], Safety Net [9].
4. Methods: Composed method [14], Execute around method [8].
5. Programming: Better for-loop construct [13], [4], Return Boolean evaluations [13], No Null beyond method scope [13].

After that, from each category we selected 2 idioms to be used in the experiment. The selection was done on the basis of idiom recognisability and the easiness of implementation in our sample source code.

3 Source Code Maintainability

Software quality can be defined as the degree to which a system, component, or process meets specified requirements or the degree to which a system, component, or process meets customer or user needs or expectations [10]. This definition is very general and cannot directly be a base for software quality evaluation. To focus ones attention to a specific aspect, software quality is defined in terms of quality attributes

or characteristics. Quality characteristics can be further subdivided into sub-characteristics forming a quality model. Characteristics and sub-characteristics can form a tree or a graph.

The general definition of maintainability is similar across different quality models (e.g. ISO 9123, ISO 25010) but each of them introduces different sub-characteristics. Sub-characteristics of maintainability may influence each other. Some suggestions about the existing relationships between them can be found e.g. in [12]: modularity and analysability can influence modifiability or modifiability is a combination of changeability and stability. Other relationships can be derived from definitions of sub-characteristics.

After careful investigation we found following sub-characteristcis of maintainability interesting in the context of research question:

- Analysability – the capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified [11].
- Understandability – the extent to which the purpose of source code is clear to a reader [11].
- Modifiability – the degree to which a product can be effectively and efficiently modified without introducing defects or degrading performance [12].
- Stability – the capability of the software product to avoid unexpected effects from modifications of the software [11].
- Testability – the ease with which test criteria can be established for a system or component and tests can be performed to determine whether those criteria have been met [12].

4 Related Works

We could not find any reports from experiments evaluating the impact of programming idioms on the source code maintainability but we found some works investigating the influence of using design patterns [17], [18], object oriented programming [5], [1] or continuous code refactoring [19]. Although they differ in details (number of programs, maintainability tasks, groups, dependent and independent variables), they use similar experimental procedures and had similar purpose – to check if the particular source code feature or implementation technique has any influence (positive/negative) on the program maintainability.

Studies were mainly conducted in an academic environment – students or graduates of Computer Science participated in the experiments. Number of participants varied from 12 [19] to 58 [1]. They were divided into groups either randomly or by using random block assignment method. Each group performed the same tasks but on different source code version. Every program used in the experiment had two source code versions based on the same requirement specification, each one implemented with different software engineering technique.

Maintenance tasks were mainly concerned with adding new or modifying existing functionalities (adaptive or perfective maintenance) or bugs correction (corrective

maintenance). Often, before the real experiment, a pilot study was performed to evaluate adequateness of the task list. In almost all studies maintainability tasks were performed on computers (the exception [17]). In all cases, before the experiment, a survey was performed. Participants' education, programming skills and experience was checked in order to divide them into uniform groups. In the first three experiments the second survey was performed to get a feedback from participants.

To measure the maintainability of the source code usually the following dependent variables were taken into account: time (how long participants were working on a given task), correctness (how good the solution was) and size of the solution. Version of the source code, type of the task and participants' knowledge and experience were considered as independent variables. The statistical methods were used to verify formulated hypotheses.

5 Experiment Description

5.1 General Hypotheses and Metrics

At the beginning we set up the following null (H_0) and alternate (H_1) hypotheses.

H_0: The usage of the selected Java idioms has no influence on source code maintainability as evaluated by the following sub-characteristics: understandability, analysability, modifiability, stability and testability.

H_1: The usage of the selected Java idioms has a positive influence on source code maintainability as evaluated by the following sub-characteristics: understandability, analysability, modifiability, stability and testability.

We followed the Goal Question Metric method [2] to develop an appropriate set of metrics used in experiment. Using the definitions of maintainability characteristics, we formulated questions that provide the input for measures definition.

Q1 (Analysability): Can one find (and how quickly) the code related to the problem or the requested change?

Q2 (Understandability): Can one answer the question (and how quickly) about code purpose?

Q3 (Modifiability): Can one change (and how quickly) the code according to new specification?

Q4 (Stability): Can one make (and how quickly) the change with a low risk of breaking existing features?

Q5 (Testability): Can one establish test criteria and perform tests (and how quickly)?

The most difficult step was to define the metrics addressing particular questions. We reused the metrics used by other researchers, described in the previous section, when possible. Table 1 presents the issues (maintainability sub-characteristics) together with associated metrics. Effectiveness metrics were multiplied by 1000 to avoid very small results.

Table 1. Metrics proposed for maintainability evaluation

Q#	Metrics
Q1	Analysis Task Correctness (*ATC*) – the degree to which found fragments of source code are related with a given problem/issue/change: $ATC = \begin{cases} 0, & no\ fragment\ found \\ 0.5, & at\ least\ half\ fragments\ found \\ 1, & all\ fragments\ found \end{cases}$
	Analysis Correctness (*AC*) – the sum of *ATC* for all analysis tasks: $AC = \sum_i^n ATC_i$
	Analysis Task Time (*ATT*) – time (in seconds) of analysis task completion
	Analysis Time (*AT*) – total time of analysis: $AT = \sum_i^n ATT_i$
	Analysis Efficiency Factor (*AEF*) – efficiency of analysis tasks: $AEF = 1000 * AC/AT$
Q2	Understanding Task Correctness (*UTC*) – the degree to which an answer for a question is correct; the number from the set {0, 0.5, 1} when 0 means incorrect answer, 0.5 – partially correct answer, 1 – fully correct answer
	Understandability Correctness (*UC*) – the sum of UTC for all understandability tasks: $UC = \sum_i^n UTC_i$
	Understanding Task Time (*UTT*) – time (in seconds) of understandability task completion
	Understanding Time (*UT*) – total time of understanding: $UT = \sum_i^n UTT_i$
	Understanding Efficiency Factor (*UEF*) – efficiency of understandability tasks: $UEF = 1000 * UC/UT$
Q3	Modification Task Correctness (*MTC*) – value describing if the modification is correct (1) or incorrect (0).
	Modification Correctness (*MC*) – the number of modification tasks solved correctly: $MC = \sum_i^n MTC_i$
	Modification Task Time (*MTT*) – time (in seconds) of modification task completion.
	Modification Time (*MT*) – total time of modification tasks: $MT = \sum_i^n MTT_i$
	Modification Efficiency Factor (*MEF*) – efficiency of modification tasks efficiency: $MEF = 1000 * MC/MT$
Q4	Modification Instability (*MI*) – the percentage of test cases failed after modification assuming that before modification 0 test cases failed: $MI = \left\lvert NFT/NT \right\rvert * 100\%$, where: *NFT* – the number of failed test cases, *NT* – the number of all test cases
Q5	Testing Task Correctness (*TTC*) – the degree to which a defect was found (0 – defect not found, 0,5 – defect partially found, 1 – defect found)
	Testing Correctness (*TC*) – the sum of *TTC* for all testing tasks: $TC = \sum_i^n TTC_i$
	Testing Task Time (*TTT*) - time (in seconds) of testing task completion
	Testing Time (*TT*) – total time of testing: $TT = \sum_i^n TTT_i$
	Testing Efficiency Factor (*TEF*) – efficiency of testing tasks: $TEF = 1000 * TC/TT$

5.2 Pilot Study

One week before the experiment, a pilot study was performed. The purpose of the study was to: (1) Check if tasks are clear and understandable. (2) Reject too difficult or too simple tasks. (3) Determine time needed for each task. (4) Verify correctness of the Sorter program used in the experiment.

Five persons took part in the pilot study which was performed in the sequential way – each next participant worked on the task list corrected after the comments of the previous one. The pilot study showed that there were too many tasks on the initial lists. We assumed that each experiment shouldn't take longer than 1h 45 min.

5.3 Experimental Subjects

The study was conducted at Wroclaw University of Technology (faculty: Computer Science and Management, specialization: Computer Science) in June 2012. Fifty-five students took part in the experiment. They were divided into two main groups depending on maintainability task type. All the students had little knowledge about Java idioms.

Within each main group we had to distinguish two smaller subgroups – working respectively on the source code implemented with and without selected Java idioms. In the division process random block assignment method was used. To assign a student to a group we used so called *skillindex* which was calculated on the base of average grade of selected courses associated with programming skills, current study year and student's commercial experience.

5.4 Experimental Tasks

Experiment participants, depending on the group they were assigned to, had to solve a series of tasks. These tasks were associated either with understandability, analysability, modifiability, and stability of the source code ($List_1$) or with the code testability ($List_2$). Examples of tasks are presented later in this subchapter.

For each task there was a solution time limit set equal to 15 minutes – the time limit was agreed upon after the pilot study as the maximum task solution time of the slowest pilot study attendant. Tasks had to be performed in specific order – coming back to the previous questions was impossible. Unfortunately, it was impossible to assign all sub-characteristics to every idiom. Most of the idioms were designed to be used only in a specific programming context. For example Bouncer pattern idiom was invented to ease source code understanding and doesn't have much influence on program modifiability.

The main program (Sorter) used in the experiment had been implemented in Java by one of the authors and checked by the other. It was a console application with the following main functionalities:

- Generating sequences of random numbers.
- Loading number sequence from a text file.

- Sorting number sequence ascending or descending with a selected sorting algorithm (we used Selection Sort, Insertion Sort and Counting Sort algorithms but participants were not informed about the sorting algorithms names).
- Printing sorted result on console or saving it to a text file.

The program was implemented in 4 versions: 2 versions (with and without idioms) associated with $List_1$, and 2 versions (with idioms – 747 lines, and without idioms – 655 lines) associated with $List_2$ (evaluation of testability). The later versions were created by introducing bugs to the former versions with idioms (726 lines) and without them (634 lines).

To properly solve understandability tasks a participant had to understand selected parts of the source code and answer some questions about them. Attendants could get 0 (wrong answer), 0.5 (partially correct answer) or 1 (fully correct answer) point for a multi-part question. Below is an example of understandability task:

Find and paste constructors of NumberGenerator class, which can be used to generate: (A) decreasing numbers from (0, 1) interval, (B) decreasing integer numbers.

Tasks checking analysability of the source code required finding parts of the source code which perform specific functions or parts that need to be changed to achieve specific program functionalities. Answers were assessed in the range from 0 to 1 according to the number of found source code parts. An example of the analysability task is presented below.

Find all parts (lines) of the source code that need to be modified to achieve the following functionality: While returning the result on console and to a file we want to: (A) Print 15 numbers in each line (instead of 10), (B) Separate numbers with comma (instead of semicolon).

Modifiability and stability tasks were associated with analysability or understandability tasks – usually participants had to understand or find selected parts of code first (within understandability/analysability task) and then modify them (modifiability task). An example of modifiability task (being a continuation of analysability task) is given here.

Change the way in which the sorting results are printed on console and to the file to achieve the following functionality: While returning the result on console and to the file we want to: (A) Print 15 numbers in each line (instead of 10), (B) Separate numbers with comma (instead of semicolon).

Even if the example looks trivial, in many cases it happened that the modification was done in an unexpected way, e.g. instead of changing a constant from 10 to 15 one replaced the upper border in the loop statement.

Testability tasks evaluated an ease of diagnosing bugs in the source code – participants, having test scenarios, were required to find and correct faulty parts. Testability was understood here in a broader sense than is suggested by testability definition, as a combination of testing and modification activities. During the pilot experiment we observed that participants being asked for finding bugs at once started to correct them. It was natural way of working with the code.

6 Obtained Results and Data Analysis

6.1 Data Analysis

We started with outliers detection. We used Tukey test with Tukey hinge distance factor equal to 1.5 (for outliers) and 3 (for extreme outliers). The borders were set to the 1st and 3rd quartile. Every case of outlier was separately considered, and – depending on its nature – eliminated or not from further investigation.

We found one outlier (very low value) for *AC* (analysability correctness) metric in the group without idioms. The person had very low index value that means lack of experience in programming. We decided to eliminate this value from further consideration. We found also two extreme outliers for *MI* (modification instability) metric in both groups (with and without idioms), but both were kept because in the group without idioms that was the only value different from 0.

Next, we performed basic statistical analysis (average, median, etc. – see Table 2). For understandability and analisability metrics the collected results were better for the source code with idioms – we observed lower (for time), and higher (for correctness and efficiency) average and median metric values. The opposite results were obtained for modifiability measures: *MC*, *MT*, and *MEF*, for which the source code without idioms was higher ranked. It could be explained by the fact that students had low experience with using idioms in practice; some idioms are not very common and could cause troubles during modification.

The attention should be paid to *MI* values. This metric was evaluated with the use of both automatic (10) and manual (18) test cases, representing regression tests, checking the basic functionalities of Sorter program. All tests were done from black-box perspective. We checked here if modification had any negative effects to existing code behaviour. A program of only one person from the group without idioms failed some tests, contrary to the group with idioms (programs of 7 people failed). It could happen that a modification was evaluated as incorrect from white-box perspective (low value of *MC* metric), and at the same time the person could be given *MI* equal to 0 (all tests passed). If all modifications were done correctly from white-box perspective, MI was always equal to 0.

We tested H_0 and H_1 for all of the metrics from Table 1. All hypotheses were checked with α=0.1 (one-tailed).

Firstly, we checked with Shapiro-Wilk test if obtained values for analysability measures had normal distribution – it was true only for *AT* measure. So we could use Levene's test to check the equality of variances of *AT* values, and then Student's t-test to check our hypothesis. The obtained Student's t-test value ($p\text{-value}_{AT}$=0.718) showed that we can't reject the null H_0^{AT} hypothesis. To check the other hypothesis associated with the analysability we used the Mann-Whitney U test, but the results also did not allow us to reject the null hypotheses H_0^{AC}, H_0^{AEF}.

We followed the same procedure for understandability sub-characteristic. The results of Shapiro-Wilk test showed that values of *UT* measure had normal distribution, and values of *UC*, and *UEF* had not. We checked our hypotheses about understandability with t-Student's test (p-value=0.98) and U Mann-Whitney's test ($p\text{-value}_{UC}$=0.98, $p\text{-value}_{UEF}$=0.61) appropriately, but with no significant results. None of null hypotheses: H_0^{UC}, H_0^{UT}, $H_0^{UE}{}_F$ can be rejected.

Table 2. Basic statistics for all metrics (WI – without idioms, I – with idioms)

Metric	Ver.	No	Avg.	Median	Min	Max	Std Dev
AC	WI	12	3.333	3.5	2	4	0.615
	I	15	3.367	4	2	4	0.743
AT	WI	12	831.333	797.5	392	1384	304.057
	I	15	791.467	668	459	1237	263.241
AEF	WI	12	4.539	3.745	2.320	8.929	2.309
	I	15	4.773	4.532	1.659	8.715	1.957
UC	WI	13	4.295	4.5	2.5	5	0.717
	I	15	4.300	4.667	3	5	0.696
UT	WI	13	1209.923	1244	743	1698	271.730
	I	15	1165.467	1084	787	1806	296.326
UEF	WI	13	3.717	3.469	1.932	5.543	1.041
	I	15	3.980	4.305	2.007	6.227	1.348
MC	WI	13	2.692	3	1	4	1.032
	I	15	2.333	3	0	4	1.345
MT	WI	13	765.692	782	392	1156	202.852
	I	15	899.933	830	618	1414	253.560
MEF	WI	13	3.889	3.534	1.073	7.653	2.099
	I	15	2.883	2.890	0	6.472	1.869
MI	WI	13	0.008	0	0	0.107	0.030
	I	15	0.057	0	0	0.429	0.110
TC	WI	13	2.385	2	0	5	1.660
	I	14	4.036	4.5	2	5	1.135
TT	WI	13	2818.846	2922	2087	4453	769.269
	I	14	2934.071	3034.5	1598	3978	661.821
TEF	WI	13	0.959	0.674	0	2.276	0.805
	I	14	1.503	1.509	0.683	3.129	0.716

Values of *MC*, *MT*, and *MEF* had normal distribution that enabled us to use Student's t-test. However, the results ($p\text{-value}_{MC}$=0.44, $p\text{-value}_{MT}$=0.14, $p\text{-value}_{MEF}$=0.19) did not confirm that the observed differences between code with and without idioms are statistically significant.

Values of *MI* did not have normal distribution. The value of U test (U=61.5, p-value=0.102) was very close to the critical value 61, but still inside the critical region, so also in this case the results were not statistically significant.

Testability was represented by three derived measures: *TT*, *TC*, and *TEF* from which only values of *TEF* had normal distribution. The U Mann-Whitney's test did not show any significant results for *TT* values. However, the value U=37 was lower than critical value 56 for *TC* ($p\text{-value}_{TC}$=0.009), that means that we can reject the null hypothesis H_0^{TC} and confirm that there is a significant difference between an average testing correctness of the code with idioms and without them. So we can say that code testing (including fault correction) with idioms is easier.

The Student's t-test showed also a significant result for *TEF* measure (t=-1.859, p-value$_{TEF}$=0.075) what allowed us to reject the null hypothesis H_0^{TEC} and confirmed that the code with idioms can be more efficiently tested (number of diagnosed and corrected bugs in the same time is higher).

6.2 Internal Validity

Internal validity is a degree to which the observed effects depend only on the experimental variables. Due to a small group sizes, main threats for the internal validity in our experiment were intergroup differences between the subjects in terms of their programming knowledge and abilities. We tried to balance the groups by using random block assignment method based on the participant's skill index which was calculated from grades, study year and work experience. Unfortunately, grades, study year and experience not always need to indicate subject's programming abilities.

Another threat for the internal validity is that subjects could cheat and copy from each other – sometimes participants sitting next to each other had the same task list. We reduced this problem by using time limits and supervisors. Besides, most of the questions were open-ended (especially modifiability questions).

Finally, not all subjects could be motivated for solving tasks. Some of them could answer the questions as fast as possible without thinking. We tried to reduce this problem by motivating the students – they got extra points in the course they were enrolled in.

6.3 External Validity

External validity is a degree to which the results can be generalized and transferred to other situations. We have to consider several differences between the experimental situation and real-world maintenance which can make impossible to generalize experiment results.

Usually programmers maintain the source code they know (at least to some extent). Participants of our study saw the source code for the first time during the experiment, but on the other side the program was of small size and low complexity.

Sometimes the same person implements and maintains the program – we did not consider such cases.

In a real life programs and maintenance tasks are much more difficult, problems have different nature and come from many different sources. But, at the same time "real" programmers usually are more experienced.

Time pressure – in software companies people, who are responsible for source code maintenance have usually more time for their tasks (but tasks are more difficult).

In our study we used only selected set of Java idioms. We have many idioms and with every new version of Java, new idioms are introduced. We tried to select representative idioms from all groups of idioms we identified.

7 Conclusions

Idioms (programming patterns) are considered as the lowest level patterns. They are recommended as good practices by many practitioners. The goal of our investigation was to determine if the belief about positive influence of programming idioms on program maintenance is justified or if it is only a myth.

In order to answer that question, first we had to refine the notion of a programming idiom, and then to provide the classification of existing idioms. We considered only idioms with a scope limited to one class. We had also to refine the notion of software maintenance. We selected understandability, analysability, testability, modifiability and stability as primary maintainability drivers. Based on the descriptions of different experiments evaluating software maintenance, we reused (when possible) and proposed some measures for checking maintainability sub-characteristics.

The experiment conducted by us partially confirmed the hypothesis stating that source code with selected Java idioms is easier to maintain than source code without idioms. We obtained statistically significant results for corrective maintenance, i.e. activities containing finding (testability) and correcting (modifiability) faults. For other maintainability sub-characteristics we did not obtain significant results, however almost in all cases (except modifiability considered separately) the basic statistics (average and median) were better for the code with idioms. We suppose that in the long run the results would accumulate to bigger differences. The worse results for modifiability resulted mainly from other then expected ways of performing modifications. We are going to repeat the experiment once again (with small modifications) to check the previously obtained results.

Finally, we can conclude that – in general – using Java programming idioms is a good practice that can be recommended as one of the source code's quality elements. Using idioms seems to be especially beneficial for good Java programmers. People with less programming knowledge obtained in the experiment better results for source code without idioms. This observation is a little bit surprising and needs further investigation, e.g. by performing other experiments with experts and novice in the area of programming patterns. Idioms can also serve as means of Java programming education.

References

1. Bandi, R.K., Vaishavi, V.K., Turk, D.E.: Predicting Maintenance Performance Using Object-Oriented Design Complexity Metrics. IEEE Trans. Softw. Eng. 29(1), 77–87 (2003)
2. Basili, V.R., Caldiera, G., Rombach, D.H.: The Goal Question Metric Approach. Encyclopedia of Software Engineering. Wiley (1994)
3. Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P., Stal, M.: Pattern-oriented software architecture. In: A System of Patterns, vol. 1, pp. 345–357. John Wiley & Sons, Chichester (1996)
4. Bloch, J.: Effective Java, 2nd edn., pp. 5–27, pp. 209–240. Pearson (2008)
5. Daly, D., Brooks, A., Miller, J., Roper, M., Wood, M.: The effect of inheritance on the maintainability of object-oriented software: an empirical study. In: Proc. International Conference on Software Maintenance, Washington, pp. 20–29 (1995)

6. Deissenbock, F.: Continuous Quality Control of Long-Lived Software Systems, der Technischen Univ. Munchen, p. 11 (2009), http://mediatum2.ub.tum.de/doc/737380/737380.pdf
7. Goldschmidt, T., Reussner, R., Jochen, W.: A case study Evaluation of Maintainability and Performance of Persistency Techniques. In: Proc. The 30th International Conference on Software Engineering, New York, pp. 401–410 (2008)
8. Haase, A.: Java Idioms – Code Blocks and Control Flow. In: Rüping, A., Eckstein, J., Schwanninger, C. (eds.) Proc. The 6th European Conference on Pattern Languages of Programs (EuroPLoP 2001), pp. 227–250. UVK – Universitaetsverlag Konstanz (2002)
9. Haase, A.: Java Idioms – Exception Handling. In: O'Callaghan, A., Eckstein, J., Schwanninger, C. (eds.) Proc. The 7th European Conference on Pattern Languages of Programs (EuroPLoP 2002), pp. 41–70. UVK – Universitaetsverlag Konstanz (2003)
10. IEEE Computer Society 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology (1990), doi:10.1109/IEEESTD/1990/101064
11. ISO/IEC 9126-1 Software engineering – Product quality. Part I – Quality Model (2001)
12. ISO/IEC 25010. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (2011)
13. Java idioms (2005), http://c2.com/ppr/wiki/JavaIdioms/JavaIdioms.html
14. Kerievsky, J.: Refactoring to patterns. Addison-Wesley Professional (2004)
15. Martin, R.: Clean Code: A Handbook of Agile Software Craftsmanship, pp. 7–11. Prentice Hall PTR, Upper Saddle River (2008)
16. Pigoski, T.: Practical Software Maintenance: Best Practicies for Managing Your Software Investment, p. 51. John Wiley & Sons, New York (1996)
17. Prechelt, L., Unger, B., Tichy, W.F., Brössler, P.P., Volta, L.G.: A controlled experiment in maintenance: comparing design patterns to simpler solutions. Soft. Engineering 27(12), 1134–1144 (2001)
18. Tichy, W., Sjøberg, D.I.K., Arisholm, E., Rombach, D.: A Controlled Experiment Comparing the Maintainability of Programs Designed with and without Design Patterns – A Replication in a Real Programming Environment. Empir. Soft. Engineering 9(3), 149–195 (2004)
19. Wilking, D., Kahn, U.F., Kowalewski, S.: An Empirical Evaluation of Refactoring. e-Informatica Soft. Engineering J. 1(1), 27–42 (2007)

Quantification of Temporal Fault Trees Based on Fuzzy Set Theory

Sohag Kabir, Ernest Edifor, Martin Walker, and Neil Gordon

Department of Computer Science, University of Hull, Hull, UK
{s.kabir@2012.,e.e.edifor@2007.,martin.walker@,
n.a.gordon@}hull.ac.uk

Abstract. Fault tree analysis (FTA) has been modified in different ways to make it capable of performing quantitative and qualitative safety analysis with temporal gates, thereby overcoming its limitation in capturing sequential failure behaviour. However, for many systems, it is often very difficult to have exact failure rates of components due to increased complexity of systems, scarcity of necessary statistical data etc. To overcome this problem, this paper presents a methodology based on fuzzy set theory to quantify temporal fault trees. This makes the imprecision in available failure data more explicit and helps to obtain a range of most probable values for the top event probability.

Keywords: Dependability Analysis, Fault Tree Analysis, Fuzzy Logic, Uncertainty analysis, Temporal Fault Trees.

1 Introduction

FTA is a widely used method for evaluating system reliability of safety-critical systems, and supports both qualitative as well as quantitative analysis. Fault trees show logical connections between faults and their causes [1] and thus make it possible to understand how combinations of failures of different components can lead to system failure. After construction of a fault tree, qualitative analysis is performed using Boolean logic by reducing it to minimal cut sets (MCSs), which show the smallest combinations of failure events that are necessary and sufficient to cause the top event. Quantitative analysis of a fault tree can estimate the probability of the top event occurring from the given failure rates of basic failure modes of the system [1].

Even though FTA is a powerful technique widely used in reliability engineering, conventional fault tree analysis has some limitations, e.g. in expressing time- or sequence-dependent dynamic behaviour [2–4] or in handling uncertainties and integrating human error in failure logic [5]. FTA has gone through different modifications to overcome these limitations, e.g. one recent modification is Pandora, which extends fault trees with temporal gates and provides temporal laws to allow qualitative analysis of dynamic systems [6]. Pandora can be used to determine the minimal cut sequences (MCSQs) that cause the top event.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_24,

The outcome of any quantitative analysis largely depends on the accuracy of the failure rates used in the analysis. In conventional FTA, failure rates of components are typically considered to be constant. However, for many complex systems, it is often very difficult to estimate a precise failure rate of components from past occurrences due to lack of knowledge, scarcity of statistical data, and changes in operating environments of the systems etc. [5, 7]. This situation is especially relevant in the early design stages because at that time we may have to consider failure rates of new or undetermined components which have no available quantitative failure data, and thus precise failure rates could not possibly be known. Therefore, human judgment by linguistic expressions, such as 'very low, low, high, very high' can be used to define failure rates. In order to allow the conventional FTA to use linguistic variables and capture uncertainty, different modifications and improvements based on fuzzy logic have been proposed by different researchers [5, 7–11]. Fuzzy Logic is a branch of mathematics that deals with linguistic variables and provides an efficient way to draw conclusions from imprecise and vague information [12].

Recently, attempts have been made to quantify temporal gates in Pandora temporal fault trees [13, 14], but all of them are based on constant failure rates. These approaches do not consider inclusion of a degree of uncertainty in the failure rates of the basic events. However, if uncertainties are left unresolved, then even the most sophisticated and well-defined quantitative model may produce misleading results [5]. Therefore, in this paper, a fuzzy set theory based methodology is introduced to quantify Pandora temporal fault trees, overcoming the limitations in handling uncertainties in failure probabilities and allowing the use of linguistic variables. The failure rates of basic events / components are defined as fuzzy numbers, and then top events probabilities are calculated based on these numbers.

2 Preliminaries on Fuzzy Set Theory

2.1 Fuzzy Numbers and Fuzzy Sets

Fuzzy set theory has been developed to deal with imprecise, vague or partially true information [15]. A fuzzy number A can be thought of as a set of real numbers where each possible value has a weight between 0 and 1 which is referred to as the degree of membership defined by a membership function. Among different forms of fuzzy numbers, triangular fuzzy number (TFN) and trapezoidal fuzzy number (TZFN) are widely used in reliability analysis. Let $x, a, b, c, d \in \mathbb{R}$, and $\mu_{\mathrm{A}}(x)$: $\mathbb{R} \rightarrow [0,1]$ represents a membership function. Then, a trapezoidal fuzzy number $A = (a, b, c, d)$ is defined by the membership function as:

$$\mu_{\mathrm{A}}(x) = \begin{cases} \frac{x-a}{b-a} & for\ a \le x \le b, \\ 1 & for\ b \le x \le c, \\ \frac{x-d}{c-d} & for\ c \le x \le d, \\ 0 & otherwise. \end{cases} \tag{1}$$

where $< b < c < d$.

A fuzzy set $\tilde{A}$ of a fuzzy number A is defined by ordered pairs, in a binary relation:

$$\tilde{\mathrm{A}} = \{(x, \mu_A(x)) \mid x \in A,\ \mu_A(x) \in [0,1]\} \tag{2}$$

where the membership function $\mu_{\mathrm{A}}(x)$ specifies the degree to which any element x in A satisfies the predefined property P. Large values of $\mu_{\mathrm{A}}(x)$ indicate a higher degree of membership.

Different methods are available to generate fuzzy numbers when no statistical data are available to estimate exact failure rates of components, e.g. 3σσ expression or expert knowledge elicitation [16]. The principle of the 3σσ method is described in [5]. The expert elicitation method has two basic forms: linguistic variables and interval values. The concept of linguistic variables is useful when little statistical data are available to estimate the failure rates of components of a system. The values of linguistic variables are words or sentences in natural languages. For example, we can consider "failure rate of component" as a linguistic variable consisting of fuzzy sets like *very low*, *low, fairly low, medium, fairly high, high, very high.* Linguistic variables play an important role in dealing with situations which are too complex or vague in nature, i.e., very difficult to describe using conventional quantitative expressions. Basic events can be assessed in a natural way and failure rates of the events can be estimated by suitable membership functions e.g., triangular or trapezoidal membership functions. Lower and upper bounds of a membership function can be obtained either from the point median value and an error factor or by direct assignment based on expert opinion.

2.2 Fuzzy Operators for Boolean Gates

Analogous to conventional FTA, the following fuzzy operators can be defined for the AND and OR gates of the temporal fault tree analysis (TFTA) [11].

AND gate fuzzy operator:

In TFTA, for all statistically independent events, the AND gate fuzzy operator is $P_{AND} = \prod_{i=1}^{n} P_i\,(t)$, where $P_i(t)(i = 1,2,3 \ldots n)$ is the failure probability of event *i* at time *t*. If the failure probability of event *i* is presented by a fuzzy number as $P_i(t) = (a_i(t), b_i(t), c_i(t), d_i(t))$, then the AND gate fuzzy operator is:

$$P_{AND} = \prod_{i=1}^{n} P_i(t) = \left(\prod_{i=1}^{n} a_i(t), \prod_{i=1}^{n} b_i(t), \prod_{i=1}^{n} c_i(t), \prod_{i=1}^{n} d_i(t)\right) \tag{3}$$

OR gate fuzzy operator:

In TFTA, for all statistically independent events, the OR gate fuzzy operator is $P_{OR} = 1 - \prod_{i=1}^{n}(1 - P_i\,(t))$, where $P_i(t)(i = 1,2,3 \ldots n)$ is the failure probability of event *i* at time *t*. If the failure probability of event *i* is presented by a fuzzy number as $P_i(t) = (a_i(t), b_i(t), c_i(t), d_i(t))$, then the OR gate fuzzy operator is:

$$P_{OR} = 1 - \prod_{i=1}^{n}(1 - P_i\,(t))$$

$$= (\,1 - \prod_{i=1}^{n}(1 - a_i(t))\,,1\; - \prod_{i=1}^{n}(1 - b_i(t))\,,1 - \prod_{i=1}^{n}(1 - c_i(t))\,,1 - \prod_{i=1}^{n}(1 - d_i(t))) \quad (4)$$

3 Pandora Temporal Fault Tree Analysis

3.1 Pandora Temporal Gates and Logic

Pandora defines three temporal gates: Priority-AND, Priority-OR, and Simultaneous-AND [13, 14]. These gates allow analysts to represent sequences or simultaneous occurrence of events as part of a fault tree.

The Priority-AND (PAND) gate is used to represent a particular sequence of events and is defined as being true only if: 1) all input events occur; 2) input events occur in sequence from left to right; and 3) no input events occur simultaneously. The symbol '< ' is used to represent the PAND gate in logical expressions, i.e. X < Y means (X PAND Y).

The Priority-OR (POR) gate is used to indicate that one input event has priority and must occur first for the POR to be true, but does not require all other input events to occur as well. It can be used to represent trigger conditions where the occurrence of the priority event means that subsequent events may have no effect. The POR is true only if: 1) its left-most (priority) input event occurs; 2) no other input event occurs before the left-most input event; and 3) no other input event occurs at the same time as the left-most input event. The symbol ' | ' is used to represent the POR gate in logical expressions, thus X|Y means (X POR Y).

The Simultaneous-AND (SAND) gate is used to define situations where an outcome is only triggered if two or more events occur approximately simultaneously, e.g. because of a common cause, or because the events have a different effect if they occur approximately simultaneously as opposed to in a sequence. It is true only if: 1) all input events occur; and 2) all events occur at the same time. The symbol '&' is used to represent the SAND gate in logical expressions.

Note that the priority of the gates is as follows: SAND is highest, then PAND, POR, AND, and OR. Thus e.g. A+B&C<D is equivalent to A + ((B&C) <D). '+' is used here to represent OR and '.' is used to represent AND. It is also important to note that in Pandora it is not possible for an event to occur both at the same time and before/after another event, as this would be a contradiction; therefore, PAND and SAND are mutually exclusive, as are POR and SAND. Thus if X.Y is true, then exactly one of X<Y, X&Y, and Y<X must also be true. Furthermore, the structure function of a Pandora fault tree is monotonic, i.e. no event or gate can ever go from an occurred to non-occurred state [6]. In this paper, events are assumed to be non-repairable, to be statistically independent, and to have failure rates with exponential distributions — all common assumptions in FTA. Under these assumptions, the probability of two events occurring exactly at the same time is 0, therefore any MCSQs

containing SAND gates will not be considered for evaluation (as the full MCSQ would also evaluate to 0).

3.2 Fuzzy Probabilities of Temporal Gates

Fuzzy operators for PAND and POR can be derived from formulae in [13] and [17].

1. Fuzzy probability of the PAND gate

In a minimal cut sequence (MCSQ), if there are *N* statistically independent input events in a PAND gate and they occur sequentially, i.e., event 1 occurs first, then event 2,… *N*–1, and finally event *N*, then the probability of that PAND gate can be defined as:

$$P_{PAND} = \prod_{i=1}^{N} \lambda_i \sum_{k=0}^{N} \left[\frac{e^{(u_k t)}}{\prod_{\substack{j=0 \\ j \neq k}}^{N} (u_k - u_j)} \right] \tag{5}$$

where $u_0 = 0$ and $u_m = -\sum_{j=1}^{m} \lambda_i$ for $m > 0$.

If the failure rate of event *i* is represented by a fuzzy number as $\lambda_i = (a_i, b_i, c_i, d_i)$, then the fuzzy probability of the PAND gate expression is:

$$P_{PAND} = \left[\prod_{i=1}^{N} a_i \sum_{k=0}^{N} \left[\frac{e^{(u_k t)}}{\prod_{\substack{j=0 \\ j \neq k}}^{N} (u_k - u_j)} \right], \prod_{i=1}^{N} b_i \sum_{k=0}^{N} \left[\frac{e^{(u_k t)}}{\prod_{\substack{j=0 \\ j \neq k}}^{N} (u_k - u_j)} \right], \right.$$

$$\left. \prod_{i=1}^{N} c_i \sum_{k=0}^{N} \left[\frac{e^{(u_k t)}}{\prod_{\substack{j=0 \\ j \neq k}}^{N} (u_k - u_j)} \right], \prod_{i=1}^{N} d_i \sum_{k=0}^{N} \left[\frac{e^{(u_k t)}}{\prod_{\substack{j=0 \\ j \neq k}}^{N} (u_k - u_j)} \right] \right] \tag{6}$$

If there are 2 input events in the PAND gate, then according to [17], equation (6) reduces to:

$$P_{PAND} = \frac{\lambda_2}{(\lambda_1 + \lambda_2)} - e^{(-\lambda_1 t)} + \frac{\lambda_1}{(\lambda_1 + \lambda_2)} e^{[-(\lambda_1 + \lambda_2) t]} \tag{7}$$

2. Fuzzy probability of the POR gate

For any minimal cut sequence of *N* statistically independent events in a POR gate with the expression $E_1|E_2|\ldots|E_{N-1}|E_N$, and failure rates $\lambda_1, \lambda_2, \ldots, \lambda_{N-1}, \lambda_N$ respectively, then the probability of the POR gate can be defined as:

$$P_{POR} = \frac{\lambda_1 \left(1 - \left(e^{-\left(\sum_{i=1}^{N} \lambda_i \right) t} \right) \right)}{\sum_{i=1}^{N} \lambda_i} \tag{8}$$

If the failure rate of event *i* is represented by a fuzzy number as $\lambda_i = (a_i, b_i, c_i, d_i)$, then the fuzzy probability of that POR gate expression is:

$$P_{POR} = \left[\frac{a_1\left(1-\left(e^{-\left(\sum_{i=1}^{N} a_i\right)t}\right)\right)}{\sum_{i=1}^{N} a_i}, \frac{b_1\left(1-\left(e^{-\left(\sum_{i=1}^{N} b_i\right)t}\right)\right)}{\sum_{i=1}^{N} b_i}, \frac{c_1\left(1-\left(e^{-\left(\sum_{i=1}^{N} c_i\right)t}\right)\right)}{\sum_{i=1}^{N} c_i}, \right.$$

$$\left. \frac{d_1\left(1-\left(e^{-\left(\sum_{i=1}^{N} d_i\right)t}\right)\right)}{\sum_{i=1}^{N} d_i} \right] \tag{9}$$

3.3 Fuzzy Top-Event Probability and Most Likely Failure Probability

Quantitative analysis of a temporal fault tree provides a way to estimate the probability of the top event occurring from the given failure rates of basic components. All the basic event failure rates are considered as fuzzy numbers to minimize error due to vagueness or uncertainty in the data. The fuzzy probability of MCSQs consisting of AND, PAND and POR gates are estimated by using (3), (6) and (9) respectively. On getting the fuzzy probabilities of all MCSQs, the fuzzy top-event probability can be obtained by (4). As failure rates are considered as fuzzy numbers, all results obtained are also fuzzy numbers with a membership function. We can represent fuzzy set of fuzzy top-event probabilities $P_T \triangleq (P_{aT}, P_{bT}, P_{cT}, P_{dT})$ as:

$$\tilde{P}_T \triangleq \{(P_{aT}, \mu(P_{aT})), (P_{bT}, \mu(P_{bT})), (P_{cT}, \mu(P_{cT})), (P_{dT}, \mu(P_{dT}))\}$$

where $P_{aT}, P_{bT}, P_{cT}\ and\ P_{dT}$ are elements of the fuzzy number P_T and $\mu(P_{aT}), \mu(P_{bT}), \mu(P_{cT})$ and $\mu(P_{dT})$ are membership values of those elements.

This may be one of the intended outcomes of the TFTA, but if required, the most likely top-event (failure) probability can be obtained from a fuzzy top-event probability via defuzzification; a process of mapping values from a fuzzy domain into a crisp domain. Although several other methods also exist, the weighted average method can be used to obtain the most likely top-event probability as follows:

$$M(P_T) = \frac{P_{aT} \times \mu(P_{aT}) + P_{bT} \times \mu(P_{bT}) + P_{cT} \times \mu(P_{cT}) + P_{dT} \times \mu(P_{dT})}{\mu(P_{aT}) + \mu(P_{bT}) + \mu(P_{cT}) + \mu(P_{dT})} \tag{10}$$

4 Case Study

For the purposes of illustrating the application of fuzzy logic in quantitative temporal analysis, we use the fuel system first presented in [13], shown in Fig.1. The system is a redundant fuel distribution system for a ship. Under ordinary operation, there are two primary fuel flows, one for each engine: Pump1 delivers fuel from Tank1 to Engine1, and Pump2 delivers fuel from Tank2 to Engine2. Flowmeters monitor the rate of flow to each engine and provide sensor information to the Controller.

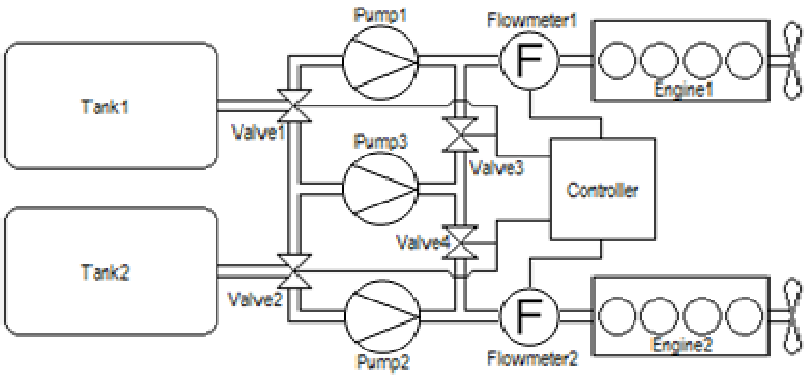


Fig. 1. Fuel distribution system

The Controller introduces dynamic behaviour to this system, allowing it to adapt to possible failures. If either flowmeter detects insufficient flow, the Controller can activate the standby Pump3 and redirects fuel flow accordingly using the valves. For example, if there is a problem with the flow to Engine1, the Controller can switch Valve1 and open Valve3 so that fuel flows from Tank1 to Engine1 via Pump3. However, Pump3 can only be used to replace either Pump1 or Pump2, but not both. A failure of both Pump1 and Pump2 will result in at least one engine being starved of fuel; for example, if Pump1 fails and Pump3 replaces it, Pump3 is then no longer available to replace Pump2 if that pump also fails. This results in degraded propulsion functionality for the vessel, as speed and maneuverability will be reduced.

Temporal gates can be used to model the dynamic behaviour in this scenario and helps to correctly capture the sequences of events that lead to failure. At the top level, the causes of omission of fuel to Engine1 can be expressed using temporal gates as follows (Engine2 is symmetrical, but with the order of events reversed):

```
O-Engine1= ((O-Pump1 | O-Pump2).O-Valve3)
           + (O-Pump2 < O-Pump1)
           + (O-Pump2 & O-Pump1)
```

Thus omission of fuel to Engine1 (`O-Engine1`) has three possible causes, depending on the sequence of events:

- If there is no fuel from Pump1 (O-Pump1), then Pump3 replaces it, as long as Pump2 has not failed first; this precondition can be represented using the POR gate. Thus in this situation, an omission of fuel can be caused by omission of fuel from both Pump1 and Pump3 (via Valve3).
- If Pump2 fails first, then Pump3 replaces it and will be unavailable to replace Pump1 if it also fails. Thus sequential failure of Pump2 and then Pump1 will lead to an omission of fuel to Engine1 (represented using the PAND gate).
- If both Pump2 and Pump1 fail at the same time (represented with the SAND gate), then Pump3 can only replace one of them. Behaviour in this situation is

non-deterministic (as Pump3 may replace either Pump1 or Pump2, but not both), and thus as a pessimistic estimation, simultaneous failure of Pump1 and Pump2 is given as a cause of failure for both engines.

After performing a qualitative analysis on this system, the resulting minimal cut sequences are as follows:

```
E1 = (P1|P2).P3 + (P1|P2).V1 + (P1|P2).V3 + (S1<P1)|P2
   + (S1&P1)|P2 + (CF<P1)|P2 + (CF&P1)|P2 + P2<P1 + P1&P2
E2 = (P2|P1).P3 + (P2|P1).V2 + (P2|P1).V4 + (S2<P2)|P1
   + (S2&P2)|P1 + (CF<P2)|P1 + (CF&P2)|P1 + P1<P2 + P1&P2
```

The failure events of MCQS are:

- P1/P2/P3 = Failure of Pump1/2/3 (e.g. blockage or mechanical failure)
- V1/V2/V3/V4 = Failure of Valve 1/2/3/4 (e.g. blockage or stuck closed)
- S1/S2 = Failure of Flowmeter1/2 (e.g. sensor readings stuck high)
- CF = Failure of the Controller

As O-Engine1 and O-Engine2 are caused by the same events in the opposite sequences, the fuzzy probability of these two top events are same. As mentioned earlier, we assume that all events are independent and the probability of two independent events occurring at the same time is effectively 0, therefore we will not consider any MCSQ consisting of SAND gate. Thus for this example system, we will not consider S1&P1|P2, CF&P1|P2 and P1&P2 during quantification of the fuzzy probability of the top event.

In this paper, we have used a trapezoidal membership function to convert basic event failure data to a trapezoidal fuzzy number. Fuzzy failure rate information for the example system is shown in Table 1. Results of the fuzzy quantitative evaluation of each minimal cut sequence of top event are shown in Tables 2 and 3 respectively. The results are obtained by considering that the system is operating at 10000 hours of its life cycle, i.e. t=10000 hours.

Table 1. Fuzzy failure rates of components for fuel system

Component	Failure rate/hour("Around") (Point median value, λ_p)	Trapezoidal representations			
		λ_A	λ_B	λ_C	λ_D
Tanks	1.5E-5	7.5E-6	1.125E-5	1.875E-5	2.25E-5
Valve1 & Valve2	1E-5	5E-6	7.5E-6	1.25E-5	1.5E-5
Valve3 & Valve4	6E-6	3E-6	4.5E-6	7.5E-6	9E-6
Pump1 & Pump2 & Pump3	3.2E-5	1.6E-5	2.4E-5	4E-5	4.8E-5
Flowmeter Sensor	2.5E-6	1.25E-6	1.875E-6	3.125E-6	3.75E-6
Controller	5E-7	2.5E-7	3.75E-7	6.25E-7	7.5E-7

Table 2. Fuzzy probability of first four MCSQs for top event O-Engine1

Failure Rate	Pr ((P1 | P2).P3)	Pr ((P1 | P2).V1)	Pr ((P1 | P2) .V3)	Pr ((S1 < P1) | P2)
λ_A	2.025E-2	6.677E-3	4.045E-3	8.696E-4
λ_B	4.067E-2	1.377E-2	8.386E-3	1.827E-3
λ_C	9.077E-2	3.235E-2	1.989E-2	4.437E-3
λ_D	1.176E-1	4.299E-2	2.656E-2	5.983E-3

Table 3. Fuzzy probability of remaining two MCSQs for top event O-Engine1

Failure Rate	Pr ((CF < P1) | P2)	Pr (P2 < P1)
λ_A	1.751E-4	1.093E-2
λ_B	3.69E-4	2.276E-2
λ_C	9.02E-4	5.434E-2
λ_D	1.22E-3	7.266E-2

Using (4) the fuzzy probability of the top event is obtained as follows:

P_T = (4.232E–2, 8.518E–2, 1.889E–1, 2.432E–1).

According to the results, the interval [8.518E–2, 1.889E–1] is the most likely range of values for the top event probability, whilst 4.232E–2 and 2.432E–1 are the lower and upper bound of the top event probability respectively. To verify the accuracy of the result the same case study was modelled in Isograph Reliability Workbench 11.0 (IRW) [18] and using the point median value of the failure rate, the top event probability was 1.497E-1, which lies within the range of most likely values obtained by the proposed method. The fuzzy top event probability can also be mapped into a single value by defuzzification using (10); if the fuzzy top event probability is as follows:

$\tilde{P}_T$ = {(4.232E–2, 0.75), (8.518E–2, 1), (1.889E–1, 1), (2.432E–1, 0.75)}.

Then using (10), the most likely top event probability is 1.395E–1 which is relatively close to the value obtained using Isograph Reliability Workbench.

5 Conclusion

In this paper, we showed how uncertainty can be incorporated in TFTA by applying fuzzy set theory to Pandora temporal fault trees. Adopting a fuzzy methodology may help to model situations where limited quantitative information is available and often only with a wide range of uncertainty. The method we present is capable of handling the linguistic variables and the imprecision of the uncertainties associated with the modelling of failures and their dependencies, and can more explicitly highlight areas of uncertainty in the data. This can lead to a more effective quantification of uncertain failure data in dynamic systems, producing more realistic and robust results that help to avoid mistaken assumptions and potential over/under estimations of system reliability. However, it is important to emphasise that the results can only be as reliable as

the input data, and the inclusion of fuzzy data cannot create accuracy where none previously existed. In future, we hope to extend this work by looking at how temporal FTA approaches like Pandora could be extended to include fuzzy logic operators, as well as to further develop practices for performing uncertainty analysis.

References

1. Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick, J., Railsback, J.: Fault tree handbook with aerospace applications. NASA office of safety and mission assurance, Washington, DC (2002).
2. Dugan, J.B., Bavuso, S.J., Boyd, M.A.: Fault Trees and Sequence Dependencies. In: Proceedings of Annual Reliability and Maintainability Symposium, pp. 286–293. IEEE, Los Angeles (1990)
3. Bruns, G., Anderson, S.: Validating Safety Models with Fault Trees. In: Górski, J. (ed.) Safecomp 1993, pp. 21–30. Springer, London (1993)
4. Walker, M., Papadopoulos, Y.: Qualitative temporal analysis: Towards a full implementation of the Fault Tree Handbook. Control Eng. Pract. 17, 1115–1125 (2009)
5. Mahmood, Y.A., Ahmadi, A., Verma, A.K., Srividya, A., Kumar, U.: Fuzzy fault tree analysis: A review of concept and application. Int. J. Syst. Assur. Eng. Manag. 4, 19–32 (2013)
6. Walker, M.: Pandora: A Logic for the Qualitative Analysis of Temporal Fault Trees. PhD Thesis, University of Hull (2009)
7. Tanaka, H., Fan, L.T., Lai, F.S., Toguchi, K.: Fault-Tree Analysis by Fuzzy Probability. IEEE Trans. Reliab. R-32, 453–457 (1983)
8. He, L., Huang, H., Zuo, M.: Fault Tree Analysis Based on Fuzzy Logic. In: Proceedings of Annual Reliability and Maintainability Symposium, pp. 77–82. IEEE, FL (2007)
9. Ferdous, R., Khan, F., Veitch, B., Amyotte, P.R.: Methodology for computer aided fuzzy fault tree analysis. Process Saf. Environ. Prot. 87, 217–226 (2009)
10. Suresh, P.V., Babar, A.K., Raj, V.V.: Uncertainty in fault tree analysis: A fuzzy approach. Fuzzy Sets Syst. 83, 135–141 (1996)
11. Mao, G., Tu, J., Du, H.: Reliability Evaluation Based on Fuzzy Fault Tree. In: IEEE 17th International Conference on Industrial Engineering and Engineering Management (IE&EM), pp. 963–966. IEEE, Xiamen (2010)
12. Zadeh, L.: Fuzzy logic. Computer (Long. Beach. Calif.) 21, 83–93 (1988)
13. Edifor, E., Walker, M., Gordon, N.: Quantification of Priority-OR Gates in Temporal Fault Trees. In: Ortmeier, F., Lipaczewski, M. (eds.) SAFECOMP 2012. LNCS, vol. 7612, pp. 99–110. Springer, Heidelberg (2012)
14. Edifor, E., Walker, M., Gordon, N.: Quantification of Simultaneous-AND Gates in Temporal Fault Trees. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 141–151. Springer, Heidelberg (2013)
15. Zadeh, L.: Fuzzy Sets. Inf. Control. 8, 338–353 (1965)
16. Cai, K.: System failure engineering and fuzzy methodology: an introductory overview. Fuzzy Sets Syst. 83, 113–133 (1996)
17. Fussell, J.B., Aber, E.F., Rahl, R.G.: On the Quantitative Analysis of Priority-AND Failure Logic. IEEE Trans. Reliab. R-25, 324–326 (1976)
18. Isograph Limited.: Reliability Workbench Version 11 User Guide (2011)

Analysis of Physical Layer Model of WLAN 802.11g Data Transmission Protocol in Wireless Networks Used by Telematic Systems

Zbigniew Kasprzyk and Mariusz Rychlicki

Warsaw University of Technology, Faculty of Transport
{zka,mry}@wt.pw.edu.pl

Abstract. WLAN 802.11g wireless network specification used in telematic systems was discussed in this paper. A model of physical layer used in WLAN 802.11g developed in Matlab/Simulink was analysed. Parameters of data transmission under different operating conditions were investigated.

Keywords: wireless communications, WLAN 802.11g specification, telematic system.

1 Introduction

Modern transport, especially in big urban agglomerations revolves around advanced technical and organisational solutions. They are intended to reduce congestion caused by excessive traffic volumes and thereby decrease environmental impact of road transport. An example solution used in small and medium transport enterprises involves email and SMS based information services [1]. Another example is comprehensive passenger information systems using wireless transmission. The passenger information system interacts directly with the passengers therefore ease of access is assured (voice messages). Furthermore, the vehicle is equipped with location devices providing passengers with information about current location and can feed that data to the management centre. First and foremost, the passenger information system, described in this article, is an advanced IT system consisting of devices enabling data transmission between the vehicle and passenger information system and the traffic control centre [2]. Electronic equipment used for those purposes are WLAN 802.11a/b/g wireless modems installed in vehicles and variable message signs assuring mobile access to reliable passenger information. It is one of the services offered by transport telematics systems. Therefore, the reliability and accuracy of wireless transmission determines how passenger information system operates as well as organisational activities reducing environmental impact of road transport. It is then justified to research wireless transmission performance under different operating conditions. Analysis of physical layer model of WLAN 802.11g data transmission protocol in wireless networks used by telematic systems (passenger information system) was

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_25,

carried out in this paper. This system is currently used by the Public Transport Authority in Mielec.

2 Model of WLAN 802.11g Physical Layer

Physical layer of the WLAN 802.11g specifications, responsible for data transmission, may be described by a model developed in Matlab/Simulink. By running a simulation of the model, both broadcasting and receipt of data via radio waves between the transmitter and receiver may be traced. The model presents implementation of the OFDM (Orthogonal Frequency Division Multiplexing) method modulated by quadrature amplitude modulation and phase modulation, and multipath fading depending on road conditions and frequency. The models simulated signal fading in order to reveal changes in data transfer rate. The model contains all the necessary components of the WLAN 802.11g [3] specification. Figure 1 shows a diagram of the model of physical layer used in WLAN 802.11g developed in Matlab/Simulink. The model shares some of its elements with model of 802.11a physical layer through adding further modules and adjusting considerably simulation parameters. Elements depicted in top part of the diagram denoted by green shading comprise the transmission scheme of the WLAN 802.11g wireless specification. Elements depicted in bottom part of the diagram denoted by red shading constitute the receiver side (figure 1).

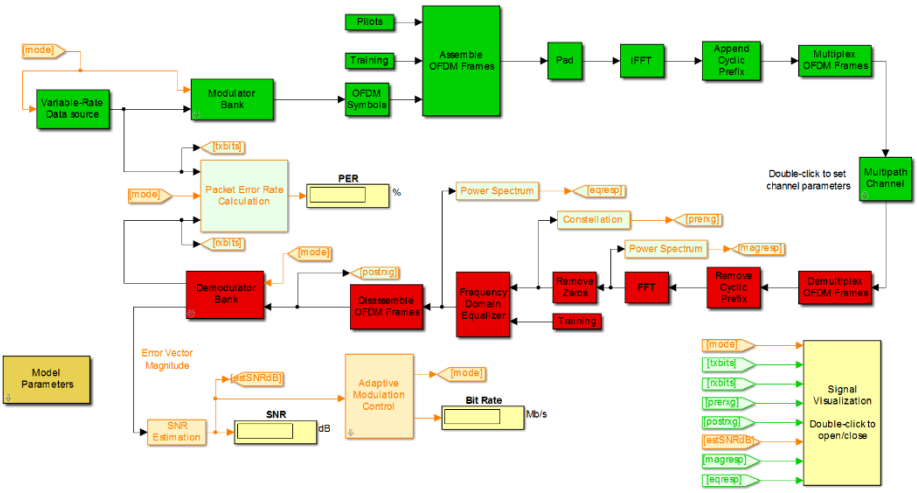


Fig. 1. Model of WLAN 802.11g physical layer

Figure 2 shows the transmitter side of WLAN 802.11g physical layer model using Orthogonal Frequency Division Multiplexing.

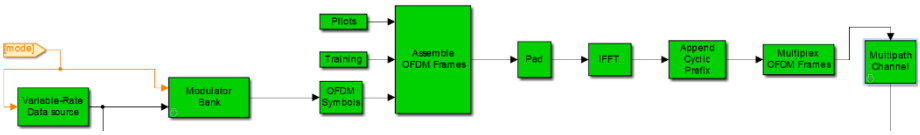


Fig. 2. Transmitter side of WLAN 802.11g physical layer model

Figure 3 shows a general diagram of the OFDM transmitter, whose functions were simulated by blocks in fig. 2.

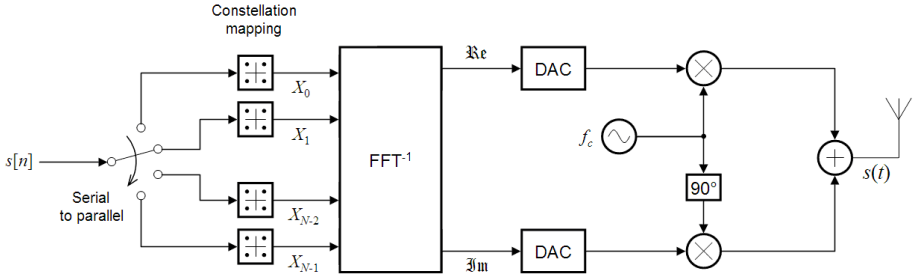


Fig. 3. General diagram of OFDM transmitter

The transmitter of the WLAN 802.11g model generates random data transfer rates varying over the course of simulation. Changes in data transfer rate of random data are controlled by the data source block. Encoding and modulation (using given scheme defined by the IEEE 802.11g standard) is carried out by the modulator bank shown in figure 4.

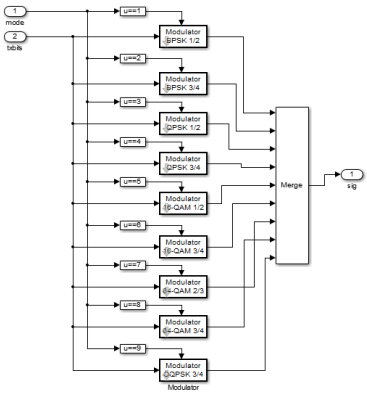


Fig. 4. Modulator bank of WLAN 802.11g transmitter

Amplitude modulation 16-QAM and 64-QAM or phase modulation QPSK, OQPSK, DQPSK and BPSK may be used in modulator bank. Cyclic Redundancy Check was used for error detection: 1/2, 2/3 and 3/4. Modulation types change

depending on bits per OFDM symbol. BPSK modulation is used for 1 bit per symbol encoding, QPSK for 2 bit per symbol encoding, 16-QAM modulation for 4 bit per character encoding, 8 bit per symbol DQPSK and 16 bit per symbol OQPSK. For instance, in 6 Mb/s version, bits are packaged into 24 bit groups for encoding and modulating. Each group is translated into 48 bit OFDM symbol, each symbol is carried by one of 48 subcarriers subject to BPSK modulation at 250 kHz (24 data bits x 250 kHz = 6 Mb/s). Similarly, in 9 Mb/s version data bits are packed into 36 bit groups. Each group is translated into 48 bit OFDM symbol, each symbol is carried subject to BPSK modulation at 250 kHz (36 data bits x 250 kHz = 9 Mb/s). That process is similar for remaining operating modes of the DQPSK and OQPSK modulators. The OFDM modulation process uses FFT^{-1} whose parameters are as follows: sampling frequency 20 mHz in 64 points. In OFDM method 48 subcarriers is used by data, 4 for pilot signals and 12 remains unused. Pilot signal is used for frame detection, estimating frequency offset of carriers and determining channel performance. OFDM modulation and frame detection through pilot signal is carried out by model blocks shown in figure 5.

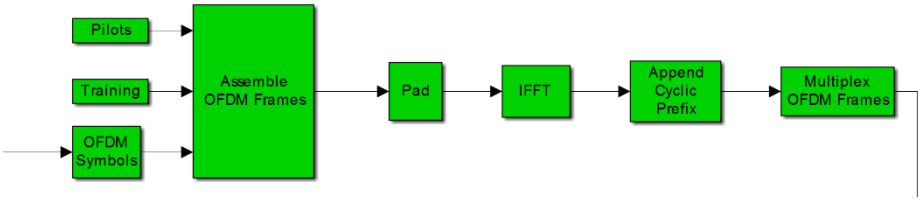


Fig. 5. Model blocks performing OFDM modulation

In OFDM modulations signal is transmitted both by signal phase and its amplitude. Fading occurs due to multipath, which distorts signal amplitude. Reference signal is transmitted by pilots thus allowing to obtain status of the channel and demodulation of neighbouring sub-channels. The OFDM signal generated using blocks illustrated in fig. 3 comprises *N* orthogonal subcarriers modulated by *N* parallel data streams. Data symbols $(d_{n,k})$ are built-up into data blocks containing *N* characters and modulated using amplitude modulation or phase-shift keying with exponential waveform $(\varphi k(t))$. After completed modulation data blocks are transmitted simultaneously as transmitter's data streams. A complete signal comprising OFDM blocks is given by the following formula [4]:

$$x(t) = \sum_{-\infty}^{\infty}\left[\sum_{k=0}^{N-1} d_{n,k\varphi k}(t - nT_d)\right] \tag{1}$$

Where: $\varphi k(t)$ describes each data subcarrier as:

$$\varphi k(t) = \begin{cases} e^{j2\pi f_k t} & t \in [0, T_d] \\ 0 & t \notin [0, T_d] \end{cases} \tag{2}$$

Where: $d_{n,k}$ transmitted signal, n_{th} time interval using k_{th} subcarrier, T_d duration of data symbol, N no of OFDM subcarriers and f_k is k^{th} subcarrier frequency derived from $f_k = f_0 + \frac{k}{T_d}, k = 0 \ldots N-1$ where f_0 is lowest available frequency.

Mutual orthogonality of subcarrier frequencies in OFDM modulation meets [4], [8]:

$$\langle s_{n1}, s_{n2} \rangle = \int_0^{T_d} s_{n1}(t), s_{n2}(t) dt = 0 \tag{3}$$

where: T_d duration of data symbol, $s_{n1}(t) = sin(n\omega t), n \epsilon \aleph$

The multipath channel block is located between transmitter and receiver side. It simulates the signal multipath process along with all related phenomena. Figure 6 shows details of the block.

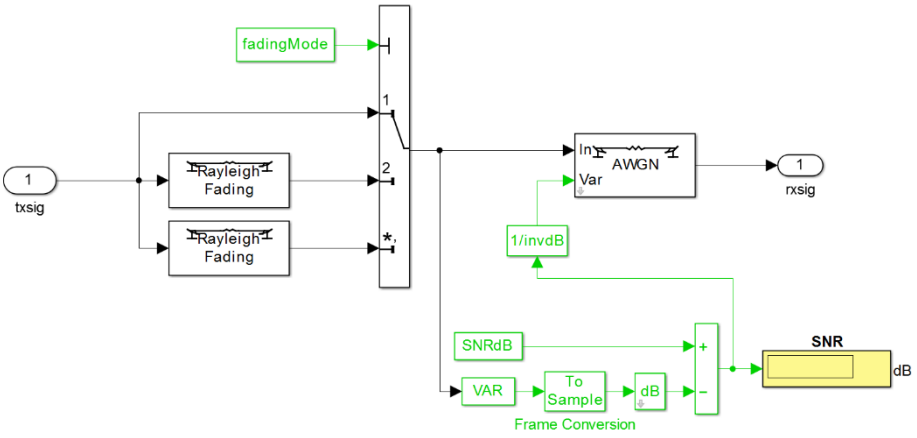


Fig. 6. Details of the WLAN 802.11g multipath channel block

It may also determine parameters of signal fading:

- fading mode:
  - no fading,
  - flat fading,
  - dispersive fading,
- maximum Doppler shift [Hz],
- Channel SNR [dB].

The most popular fading channel models are Rayleigh fading and Rician fading. Rayleigh fading model was used to simulate fading in this paper. The model assumes that channel delay and Doppler spectrum power are distinctive [5]. Let s_i denote input samples. Then output samples y_i satisfy [5]:

$$y_i = \sum_{n=-N_1}^{N_2} s_{i-n} g_n \tag{4}$$

where g_n is defined as:

$$g_n = \sum_{k=1}^{K} a_k sinc\left[\frac{\tau_k}{T_s} - n\right], -N_1 \leq \mathrm{n} \leq N_2 \tag{5}$$

where: T_s the period of input sample, τ_ks sample stream delays in fading channel, a_k sample gains in fading channel.

Relationship (6) determines the distribution function of fading probability [6]:

$$F(p) = 1 - e^{-p \cdot p} \tag{6}$$

where: $p = \frac{s_y}{s_x}$, s_y – received signal, s_x – reference signal (no fading)

Simulation of Rayleigh fading is possible by inverse relationship (6) and expressing its product in [dB] [6]:

$$\Psi = -20log\sqrt{-ln(1-x)}\ [\mathrm{dB}] \tag{7}$$

where: Ψ - fading, x - random variable of uniform distribution within interval (0;1)

Another negative phenomenon affecting wireless transmission is Doppler shift which also was simulated in this example. The Doppler shift with regards to wireless devices e.g. standard-issue GSM, is defined as follows [7]:

$$f_d = \frac{vf}{c}\ [\mathrm{dB}] \tag{8}$$

where: v - velocity of moving mobile device, f - transmission carrier frequency [Hz], c - the speed of light in vacuum [m/s]

Based on (8) with reference to velocity of moving mobile device broadcasting a signal, the maximum Doppler shift of signal transmitted by a vehicles on the move is 80 [Hz]. The signal transmitted by a mobile device carried by a pedestrian may be subject to a maximum Doppler shift of 4 [Hz]. Those values hold true for carrier frequency 900 MHz [7].

The input signal, described by formula (1) and satisfying (3) and subjected to phenomena i.e. signal multipath and Doppler effect, is received by the WLAN 802.11g physical layer receiver side. Figure 6 shows the receiver side represented by model blocks.

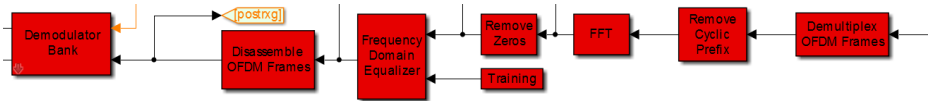


Fig. 7. Receiver side of WLAN 802.11g physical layer model

Figure 10 shows a general diagram of the OFDM receiver, whose functions were simulated by blocks in fig. 9.

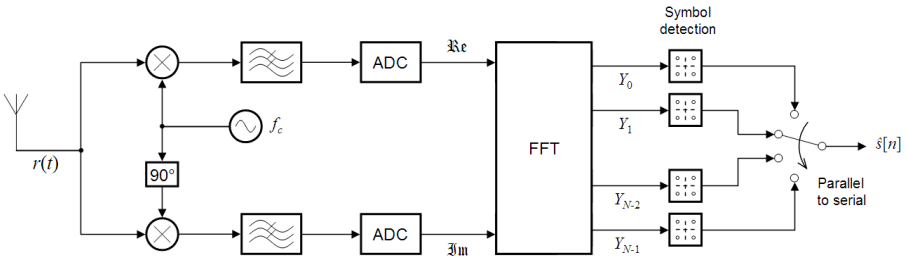


Fig. 8. General diagram of OFDM receiver

The physical layer of the WLAN 802.11g specification demultiplexes the input signal consisting of OFDM blocks. Then Fast Fourier transform FFt is calculated, signal demodulated and OFDM symbols detected. Obtained *N* parallel streams are combined into a single original bit-stream.

3 Model Simulation and Visualisation and Interpretation of Results

In order to run simulation of the Matlab/Simulink model correct input data needs to be entered first. Input data is entered by running the S-function on model level (developed in Simulink environment) or by entering them manually into text fields in model blocks. Model parameters used for simulation may be defined in Model Parameters block and the Multipath channel block (figure 1). Input data characteristic for 802.11g specification defined for simulation are as follows:

- OFDM bits per transmitted block (48 bits in 802.11g standard),
- OFDM bits and pilot bits (48 + 4 = 52 bits in 802.11g standard),
- OFDM bits per cyclic prefix (16 bits in 802.11g standard),
- Number of points where fast Fourier transform FFT occurred (64 bits in 802.11g standard),
- Hysteresis coefficient for adaptive modulation [dB],
- Depth of decision by Viterbi algorithm for decoding convolutional codes,
- SNR (signal-to-noise ratio) thresholds for different modulation schemes used in the model,
- Fading mode,
- Maximum Doppler shift [Hz],
- Channel SNR [dB],

Simulation results were observed on displays and via the Signal Visualization module producing graphic representation of results. The following output data (in graphic representation as per figure 11) were produced by the simulation:

- Packet of randomly sent binary data, tracing variable data transfer rate on the channel,
- Scatter charts depicting the signal prior and post equalization (correction). Scatter charts allow to determine modulation mode currently in use by the 802.11g protocol, because the chart represents a diagram of modulation constellations consisting of 2, 4, 16 or 64 constellation points.
- Spectrum power of received signal prior and post equalization (correction),
- SNR value estimated based on error vector magnitude,
- Bit rate,
- Bit error rate per packet,
- Packet error rate,
- Signal-to-noise ratio (SNR),
- Bit rate shows which of flowabilities defined by the specification is currently in use.

The charts above show results of model simulation using the following input data characteristic for 802.11g specification:

- OFDM bits per block = 48 bits,
- OFDM bits and pilot bits = 52 bits,
- Cyclic prefix bits = 16 bits,
- Implementation of FFT algorithm = 64 points,
- Maximum Doppler shift = 180 [Hz],
- Flat fading,
- Change in signal to noise ratio: 10, 20, 40 [dB] respectively

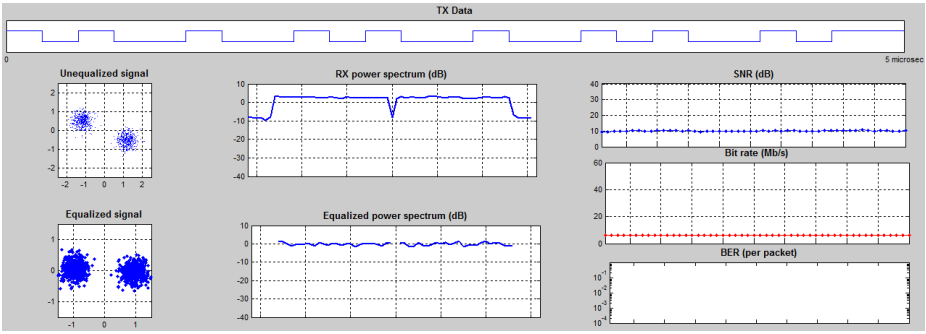


Fig. 9a. Simulation results given SNR = 10 [dB], 6 Mb/s transmission rate

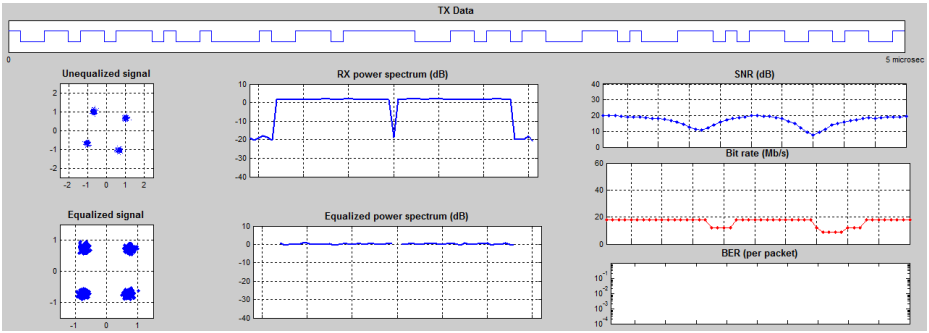


Fig. 9b. Simulation results given SNR = 20 [dB], 18 Mb/s transmission rate

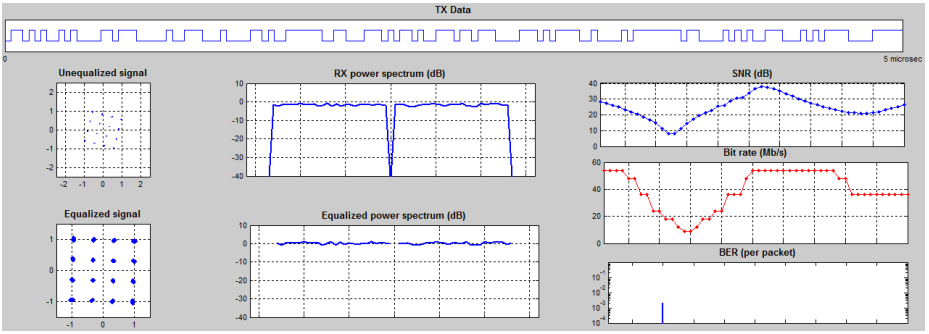


Fig. 9c. Simulation results given SNR = 40 [dB], 54 Mb/s transmission rate

Analysis of obtained results proves that fading has great impact on correct demodulation of signal in WLAN 802.11g standard, especially given high SNR values. Changes in power spectrum of received signal depend on fading parameters (figures 11a and 11c). Constellation diagrams shown in figure above are a graphic representation of digitally modulated signal. After receiving the signal, demodulator checks received symbol, which could have been distorted by either the channel or the receiver. Signal quality analysis (based on constellations during simulation) showed white noise represented by blurry constellation points and phase noise represented by arched constellation points. The orthogonal multiplexing of OFDM frequencies causes susceptibility to Doppler effect and distorts synchronization of carrier frequency. Furthermore, data transmission efficiency is lower due to cyclical prefix (figures 11b and 11c). Obtaining high data transfers reaching 54 Mb/s is difficult because of high loads and noise occurring during the transmission (figure 11c).

In further research work is expected to carry out simulations to other WLAN standards, which will include contemporary and future technologies (with particular emphasis on the use of transport telematics systems).

4 Conclusions

Simulation tests in Matlab/Simulink environment of physical layer comprising the WLAN 802.11g specification determined wireless transmission performance under different operating conditions. The WLAN 208.11g standard is susceptible to the Doppler effect. This is particularly important in case of mobile receivers and transmitters installed in public transport vehicles as in passenger information system (variable message signs). An additional disturbance is the fading mode caused by signal multipath which distorts the amplitude of radio signal. Achieving upper transfer rates reaching 54 Mb/s using the analysed specification poses substantial difficulties due to noises occurring during data transmission.

References

1. Rychlicki, M., Kasprzyk, Z.: Integracja usług poczty elektronicznej oraz sms w małych i średnich przedsiębiorstwach transportowych. Zeszyty Naukowe Politechniki Warszawskiej, Seria Transport, z. nr 92, 1230-9265. Oficyna Wydawnicza Politechniki Warszawskiej (2013)
2. Siergiejczyk, M., Paś, J., Rosiński, A.: Application of closed circuit television for highway telematics. In: Mikulski, J. (ed.) TST 2012. CCIS, vol. 329, pp. 159–165. Springer, Heidelberg (2012)
3. IEEE Std 802.11g - 2003
4. Nee, R.V., Prasad, R.: OFDM for Wireless Multimedia Communications. Artech House, Boston (2000)
5. Jeruchim, M.C., Balaban, P., Shanmugan, K.S.: Simulation of Communication Systems, 2nd edn. Kluwer Academic/Plenum, New York (2000)
6. Systemy szerokopasmowe z celowym rozpraszaniem widma, Projekt nr 2., Kontrola mocy w standardzie CDMAOne. Wydział Elektroniki i Technik Informacyjnych Politechniki Warszawskiej, Warszawa (2010)
7. http://www.mathworks.com/help/comm/ug/fading-channels.html
8. Sendra, S., Fernandez, P., Turro, C., Lloret, J.: IEEE 802.11a/b/g/n Indoor Coverage and Performance Comparison. In: 6th International Conference on Wireless and Mobile Communications, ICWMC (2010)

Web Systems Availability Assessment Considering Attacks on Service Configuration Vulnerabilities

Vyacheslav Kharchenko[1,2], Alaa Mohammed Abdul-Hadi[1], Artem Boyarchuk[1], and Yurij Ponochovny[3]

[1] National Aerospace University KhAI, Kharkiv, Ukraine
[2] Centre of Safety Infrastructure-Oriented Research and Analysis, Kharkiv, Ukraine
V.Kharchenko@khai.edu
[3] Poltava National Technical University named after Yurij Kondratyuk, Poltava, Ukraine
pnch1@rambler.ru

Abstract. The paper examines the issues of web systems assessment availability. It is defined that unavailability of web services may be caused by internal and external factors in particular server side vulnerability attacks. Three Markov's models of web system availability are developed; these models consider influence of software defects and vulnerability attacks for DNS, DHCP and Route services. Elimination of configuration vulnerabilities during system operation is considered. Conclusions about the impact of the probability of detection and elimination of vulnerabilities and the recovery rate on the web systems availability function are proposed.

Keywords: web system availability, Markov's models, attacks on vulnerability services.

1 Introduction

The successful beginning and operation of web systems is only possible in case of payback on their functioning and positive profit earning. The break-even point is reached after the start of system exploitation, and it might not be achieved at all if risk assessment was wrong. This leads to the importance of modeling the functioning of web systems based on actual cyber security risks [1-3].

Nowadays, most web services experienced the attacks of various kinds. With regard to commercial Web services, they certainly are the most attractive target for attacks [1, 4]. In such circumstances, modeling of web attacks as events that lead to their inaccessibility is in high demand. However, today the majority of the models of attacks, threats and incidents have probabilistic nature of risk assessment. Only some sources refer to the possibility of web system modeling using semi-Markov processes and Petri nets [5].

The modern web system is a complex multileveled and distributed system. It can be presented by the charts with various hierarchy levels. This paper discusses the three-component reliability block diagram of the web system (RBD). It describes the interaction of basic services: IP-address assignment (DHCP), IP routing (Route) and support the direct and inverse transformation of text URLs to IP-addresses (DNS).

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_26,

This decision is due to the fact that vulnerability subsets of mentioned services might be distinguished in line with CVE classifiers [6,7]. This allows getting estimates of the intensity of attacks and their criticality [8].

Unavailability of any of these services entails the refusal in customer service. On this basis, the RBD will include three consecutive elements, each of which corresponds the up-states of three services (fig.1).

Fig. 1. Reliability block diagram of web system

While assessing web systems availability the focus is given on Markov's models based on hardware and software failures (caused by physical and design faults correspondingly) and recoveries [9, 10]. Researches [11, 12] analyze the concept of an integrated approach of dependability as a property which combine in particular reliability, availability and information security. In [13] the possibility of the development of mathematical models that consider the unavailability of web systems in context security is proposed. Unavailability is caused by not only by software faults, but by attacks on their components as well.

The objective of this paper is to develop Markov models of web systems availability considering attacks, and to investigate the impact of input parameters of the model to the availability function. First of all, we research behavior of web systems in non-stationary modes taking into account various kinds of attacks and recovery procedures. The paper is structured as follows: the second section describes the simple Markov models of web-services without attacks (MA1) and with mechanism for restart after attack (MA2). The third section describes the MA3 model used for assessment of web-service availability considering consequent fixing of vulnerabilities after attacks. Verification results and case study of developed models are presented in the fourth and fifth sections. The last section includes the conclusions and directions of future work.

2 Availability Models of Web Systems without Attacks and with System Restart after Attack

2.1 Model MA1

We consider an ideal web system model without attacks as a basic model in which there are processes of software failures and recoveries of related network services (MA1). Resulting characteristics of such model often are used by hosting providers as the availability and uptime rate of hosting platforms.

Marked graph of states and transitions of such model is shown at the fig.2,a. It includes initial up-state S0 and down-states S1, S2 and S3. The transitions into down-states are marked with the corresponding failure rates (ladns, ladhcp and laroute). System returns into up-state after service recovery with corresponding rates mudns, mudhcp and muroute.

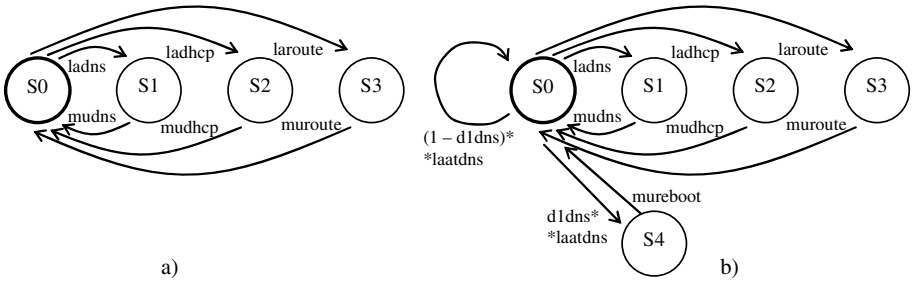


Fig. 2. Marked graphs of web system models without attacks MA1 (a) and with system restart after attack MA2 (b)

2.2 Model MA2

The second model (MA2) describes the functioning of web system during the occurrence of one attack on DNS service with system restart after successful attack without fixing of vulnerability. The graph of the model is presented at the fig.2,b. Initially web system operates with occurrence of failures and recoveries of DNS, DHCP and Route services.

Attacks on DNS service are characterized with laatdns rate и d1dns criticality. Thus the model returns to the S0 state with recovery rate (1-d1dns)*laatdns. After completing the attack (transition into S4 with d1dns*laatdns rate) the system fails. The system recovers it after restart with mureboot rate.

3 Availability Model with Detection and Elimination of Configuration Vulnerabilities

3.1 Model MA3

The model MA3 describes the operating of web system with subsequent attacks on vulnerabilities of DNS service following by their elimination without updating of software codes (ladns=const). We model the elimination of vulnerability only after its occurrence (after successful attack on it).

Two independent options after the successful attack are possible. The first option includes launch of detecting and fixing of vulnerability by transition to S5 state.

The second one might occur in case of non-detection of vulnerability and the system passes from S4 to S0 without elimination of vulnerability which may be resulted in repeating of an attack. Let us add parameter d2p for as probability rate of detecting and elimination of vulnerability.

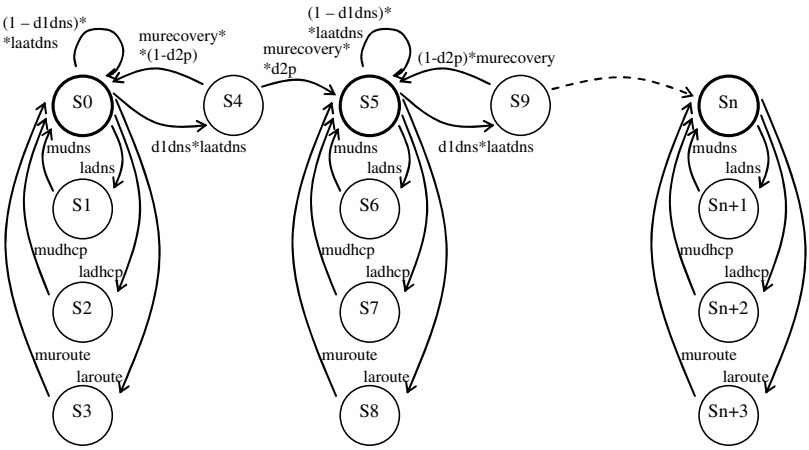


Fig. 3. Marked graph of availability model with fixing of vulnerability (MA3)

As shown at the fig.3, initially web-system operates with occurrence of faults and recovering of DNS, DHCP and Route services. After attacking the DNS service (passing to S4 state with d1dns*laatdns rate) the system is losing its operability (transition into down-state). Once occurred a vulnerability might be eliminated with d2p rate (transition from S4 into S5 with d2p*murecovery rate) or the system returns to the initial state without elimination of vulnerability with (1–d2p)*murecovery rate. After occurrence and eliminating of all vulnerabilities the system continues its operation with faults occurrence and recovering of its services (Sn…Sn+3 states).

3.2 Comparison of Availability Models Investigation Results

The results of models MA1, MA2 and MA3 were obtained by use the values of input parameters presented in table 1. Marked graph of MA3 model which accounts values of input parameters is shown at the fig.4. Technique of parameter values calculation (ladns, ladhcp, laroute and others) using information in vulnerability data bases is described in [13].

Table 1. Values of input parameters for availability models

Name	Value	Unit	Name	Value	Unit
ladns	3e-5	1/hr	laatdns	6.279e-3	1/hr
ladhcp	1.5e-5	1/hr	d1dns	0.77	
laroute	5e-4	1/hr	mureboot	0.5	1/hr
mudns	0.67	1/hr	murecovery	0.33	1/hr
mudhcp	1	1/hr	d2p	0.5	
muroute	0.33	1/hr			

The graph presented at the fig. 4 is the extended graph from one presented at the fig. 3 and amended with values of transition rates obtained from table 1 for 4 regular fragments. The graph is developed using the grPlot.m tool of Mathlab.

Solving of Kolmogorov-Chapman differential equation system were made in Mathlab system using ode15s method for the internal of [0…5000] hours. The results of comparison of developed models research are shown at the fig. 5.

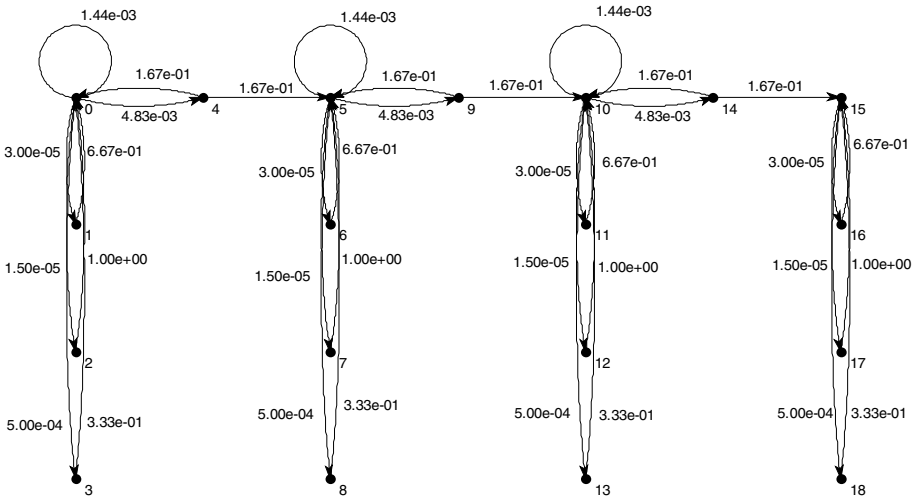


Fig. 4. Oriented graph of MA3 model for the system with three vulnerabilities

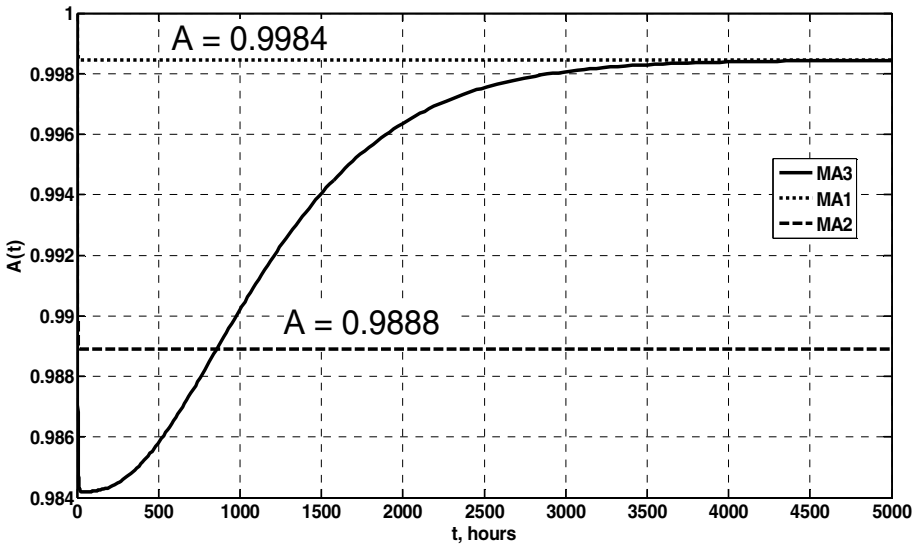


Fig. 5. Graphs of availability functions for models MA1, MA2 and MA3

Having the agreed input parameters, availability function of MA1 model reaches the stationary value A = 0.99844 during first 20 hours of operating. The value will be reached by availability function of MA3 model after 3500 hours. The availability function decreases to the stationary value of 0.9888 in case of non-fixing of vulnerabilities and recovery by system restarting only (MA2). The minimum of availability function of MA3 model settles lower than stationary value of MA2 availability function, because detecting and elimination of vulnerabilities takes more time than system restart (murecovery=0.33 < mureboot=0.5).

3.3 Influence of Input Parameters on Availability of Web Systems

The study of coherence of distinct parameters on trend and values of availability function is among the further interest. The following parameters were chosen for MA3 model (see table 2).

Table 2. Variable values input parameters of model MA3

Name	Series of values	Unit
d2p	[0 0.1 0.2 0.5 0.7 1]	
murecovery	[0.05 0.1 0.5 1 2]	1/hr

The special Mathlab cyclic constructions were designed for research on influence of mentioned parameters. Time interval was increased up to [0…15000] hours. The results of modelling are presented at the fig.6 and fig.7.

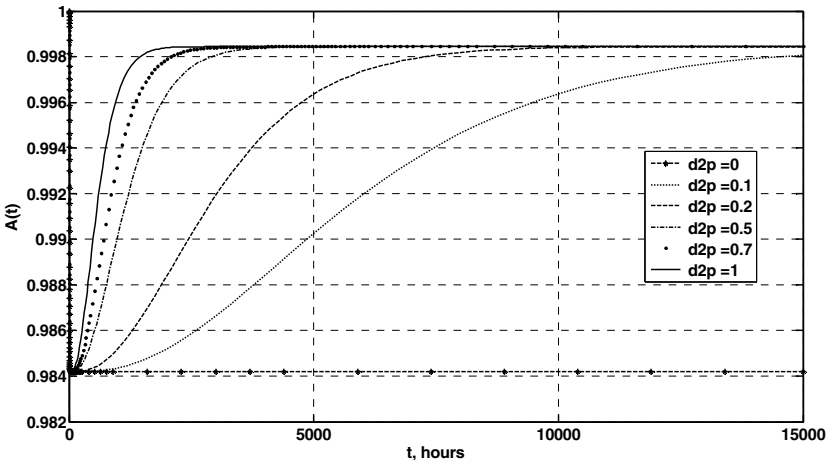


Fig. 6. Graphs of availability function of model MA3 with different values of failure repairing rate

Graphs on fig.6 illustrate the behavior of availability function caused by different values of attack detection rate d2p. Having zero rate the system degrade (shows identic results) to MA2 model with mureboot=0.33, and condition (d2p=1) results in minimal time for transforming in stationary mode.

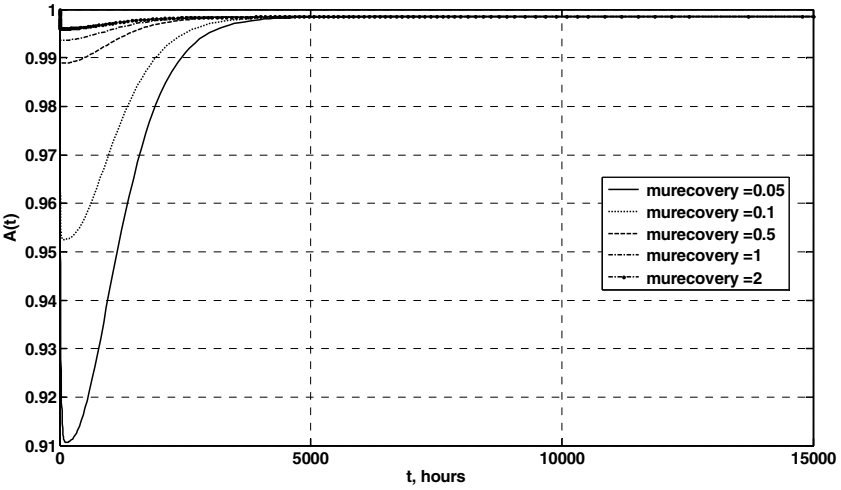


Fig. 7. Graphs of availability function of MA3 model with different values of vulnerability fixing rate and system recovery rate

Analysis of graphs confirms the correctness of the results obtained from modeling. Obviously, the value of murecovery parameter (rate of vulnerability detection and elimination jointly with system recovery rate) has an impact on the following values: minimum value of availability function, position of minimum point at the time axis, duration of transition of availability function into stationary mode.

Thus, having murecovery=0.5 (1/hour) the minimum of availability function is 0.9889 in t=25 hours; murecovery=0.05 (1/hour) leads to the minimum of availability function which equals to 0.9109 in t=117 hours. The value of murecovery parameter does not have an impact on the maximum of availability function in stationary mode A=0.99844.

4 Verification of the Models

The set of simulation models were developed using Matlab in order to make the verification of the proposed analytical models and elimination of some restrictions. As stated at the results of experiments, the obtained availability graphs for models MA1 and MA2 have common phases of exploitation: transition phase at the beginning stage of the functioning and stable phase. That's why Fig.8 shows the models MA2 and MA3 only.

The transition phase of availability function calculated by the simulation model is much longer than one calculated by analytical model, wherein the value of availability rate in stable phase are almost the same, which confirms the high trustworthiness of the analytical model.

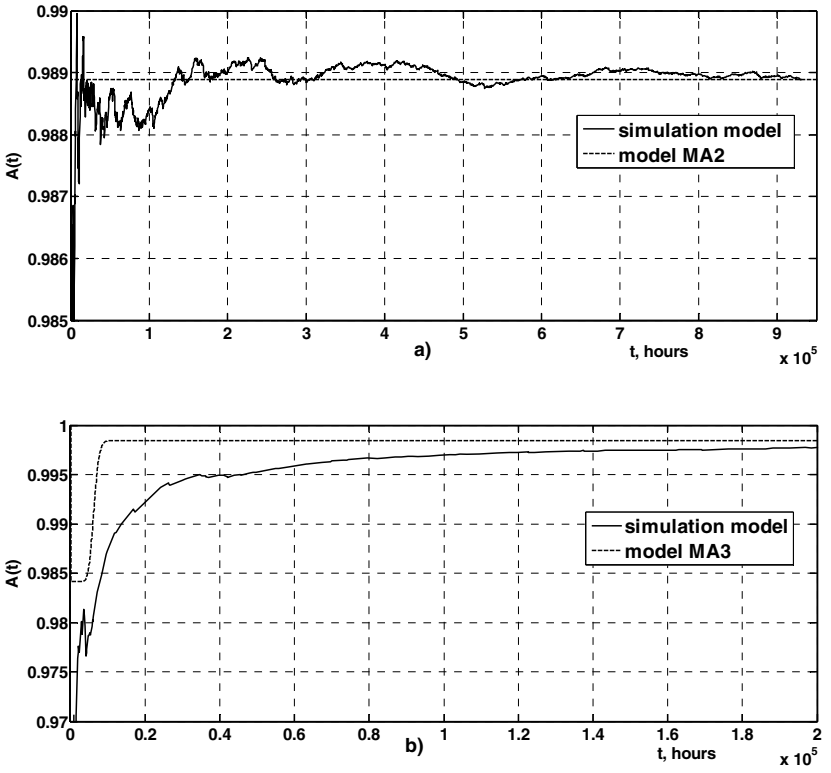


Fig. 8. Results validation of models MA2(a) and MA3 (b)

Comparison of the analytical and simulations models of the web-service with elimination of vulnerabilities (MA3) also resulted in high convergence of availability rates in stable phase (fig.8,b).

5 Case Study

The models developed at this research were successfully tested during availability assessment of e-commerce and university web-services. The decision-making system is developed on the basis of this research and aimed on ensuring of web-system availability facing attacks on its components. Its use at the design phase of web-system is shown at fig.9. The area of utilization of MA models is marked by the rectangle.

Usage of obtained results allows to increase the accuracy of evaluation of availability function and to justify choosing of probability rate of vulnerability elimination during test processes.

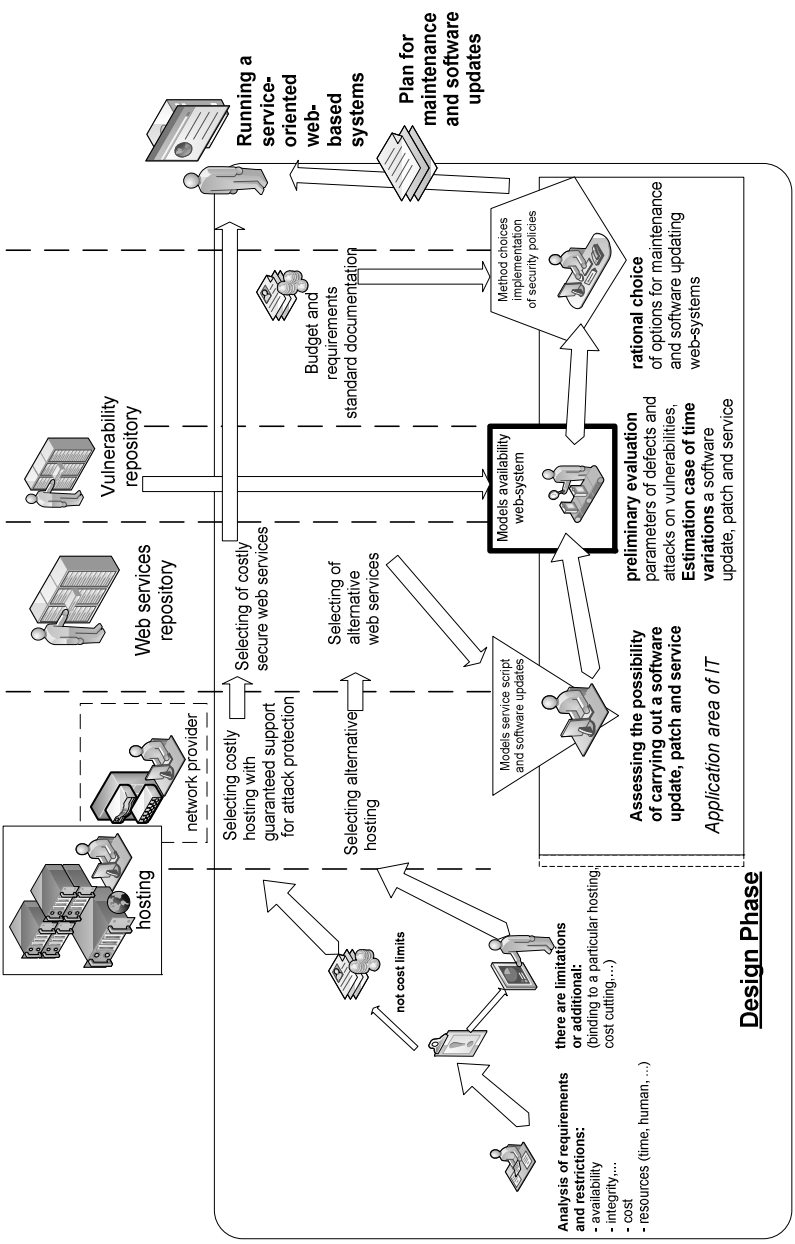


Fig. 9. IT support solutions while providing availability web system

6 Conclusions

The analysis of obtained modeling results of web system availability considering attacks on the components and vulnerability elimination shows the following issues:

a) to speed up the transition of availability function into the stationary mode it is necessary to increase the value of d2p parameter, i.e. to increase the probability rate of vulnerability detection and fixing;

b) the minimum value of availability function will rely on murecovery and mureboot parameters at the initial period of system operation (the sooner system recovers after attack the higher minimum of availability function is).

It is advisable to orient future work within the scope of development of integrated maintenance strategies of service-oriented web systems considering hardware, software tools and security policies.

Other interesting and important task is research on rationality and modes of flexible maintenance and upgrading strategies for cloud-based IT-infrastructures.

References

1. Kizza, J.M.: Guide to Computer Network Security, 2nd edn. Springer, London (2013)
2. Al-Kuwaiti, M., Kyriakopoulos, N., Hussein, S.: A comparative analysis of network dependability, fault-tolerence, reliability, security, and survivability. IEEE Communications Surveys & Tutorials 11, 106–124 (2009)
3. Hansman, S., Hunt, R.: A taxonomy of network and computer attacks. Elsevier Computers & Security 24, 31–43 (2005)
4. Security and high availability in cloud computing environments, IBM Global Technology Services Technical White Paper, IBM (2011)
5. Nicol, D.M., Sanders, W.H., Trivedi, K.S.: Model-based evaluation: from dependability to security. IEEE Transactions on Dependable and Secure Comp. 1, 48–65 (2004)
6. Recommendation X.1520. Common vulnerabilities and exposures. ITU-T, Geneva (2011)
7. Recommendation X.1521. Common vulnerability scoring system. ITU-T, Geneva (2012)
8. National Vulnerability Database, `http://nvd.nist.gov`
9. Trivedi, K.S., Vasireddy, R., Trindade, D., Nathan, S., Castro, R.: Modeling high availability systems. In: Proceedings of the 12th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2006), pp. 154–164 (2006)
10. Kim, D.S., Machida, F., Trivedi, K.S.: Availability modeling and analysis of a virtualized system. In: Proceedings of the 15th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2009), pp. 365–371 (2009)
11. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing 1, 11–33 (2004)
12. Gorbenko, A., Kharchenko, V., Tarasyuk, O., Furmanov, A.: F(I)MEA-Technique of Web Services Analysis and Dependability Ensuring. In: Butler, M., Jones, C.B., Romanovsky, A., Troubitsyna, E. (eds.) Fault-Tolerant Systems. LNCS, vol. 4157, pp. 153–167. Springer, Heidelberg (2006)
13. Abdul-Hadi, A.M., Ponochovny, Y., Kharchenko, V.: Development of basic Markov's model research availability of commercial web services. Radioelectronic and Computer Systems 5(64), 186–191 (2013)

A Recommender System Based on Content Clustering Used to Propose Forum Articles

Urszula Kużelewska[1] and Ewa Guziejko[2]

[1] Bialystok University of Technology,
15-351 Bialystok, Wiejska 45a, Poland
u.kuzelewska@pb.edu.pl
[2] Graduate of the Bialystok University of Technology
ewa.guziejko@gmail.com

Abstract. Nowadays, WWW services compete intensively to attract attention of visitors. They search new solutions to increase attractiveness of the systems and to satisfy all customer expectations. To achieve this they often offer an individual approach to each user, e.g. applying recommender systems. A recommender system is able to learn a customer's preferences and recommend products, which the user is probably interested in. The recommendations are based on similarity between registered users' activity, e.g. items, which they visited or bought.

The purpose of this paper is to find a reasonable solution to offer recommendations on internet forum. Since clustering algorithms were useful to group similar posts (according to preliminary results), they were chosen as the tool to generate recommendations. The algorithms available in Apache Mahout were used in the experiments described in this article. Finally, the recommender system has been implemented on the forum, and their effectiveness was examined, as well. The results confirmed the validity of the proposed solution.

Keywords: text clustering, recommender system, content-based recommender.

1 Introduction to Recommender Systems

Personalised recommender systems (RS) base on registered user's preferences or similarities among items. For each user, an RS generates a list of items that match user's interests. In other words, the systems filter the information that may be presented to the user based on their preferences. The selection of items on a such list can be based on product ratings, contents of user's shopping basket, as well as specific characteristics of the objects [10].

Recommendation concerns, among the others, news, music, video, content of e-learning courses, books and subject of web sites or web site navigation.

Considering a type of input data as well used methods, recommendation systems are divided into collaborative filtering (CF), knowledge-based, content-based and hybrid [10].

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_27,

Collaborative filtering techniques search similarities among users or items, however only archives of users behaviour are analysed [1]. As an example, similar users have mostly the same products in their baskets and similar items are bought by the same customers. CF methods base on the assumption, that if two users have the same opinion on the particular item, it is very likely they like similarly other items. The most important advantages of this kind of systems are: high precision, simple implementation, no additional knowledge about a domain or objects. The long list of advantages is supplemented with the following disadvantages: a problem of "cold start" for users and objects and poor scalability.

Knowledge-based approach is better for one-time users stores, e.g. selling cameras (people do not buy cameras often) [1]. The approach bases on technical attributes of the items and user's preferences, also weighted, related to the attributes. Knowledge acquirement is often realised by interaction with users. This is an approach, where the "cold start" problem does not appear and users' data are not required to store for long time, however they have to use specific techniques to gather the knowledge.

Content-based recommendations (called content-based filtering) base on attribute (characteristic) vectors of items created from text connected with the items, e.g. their description, genre, etc [10]. As an example, in case of forum articles, the item characteristics include its topic or author. The content-based algorithms recommend items, which are similar to highly rated by the user other items in past. As an example, if a user liked (rated or bought) X movie, a recommender system searched other movies, which were similar to X with regard to its genre, title, director's name or description of the story. The main advantages of content-based systems are: relatively simple implementation and independence of users. The disadvantages are: a problem of "cold start" for users and the requirement of items' features analysis.

Hybrid approach combines at least two different methods: problems in each of them are solved by strengths of the other one.

2 Clustering Methods in Recommender Systems

Clustering is a domain of data mining which had been applied in a wide range of problems, among others, in pattern recognition, image processing, statistical data analysis and knowledge discovery [6]. The aim of cluster analysis is organising a collection of patterns (usually represented as a vector of measurements, or a point in a multi-dimensional space) into clusters based on their similarity [4]. The points within one cluster are more similar to one another than to any other points from the remaining clusters.

Clustering has been the subject of research in the area of recommender systems, although it has not been widely studied yet [9]. The most often method used in memory-based collaborative filtering to identify neighbours is kNN algorithm, which requires calculating distances between an active user and the registered all ones. In contrast, clustering (in model-based collaborative filtering) reduces computation time, due to introduction of clusters models.

One of the first approaches, where clustering was used to partition users' preferences in order to increase neighbour searching efficiency is described in [12]. The authors used clusters identified in off-line mode by k-means instead of on-line neighbourhood calculated by kNN method. As a similarity measure they used Pearson correlation. Finally, except time efficiency, quality of predictions was increased.

One of the recent examples of clustering application in recommendations is [5], where k-means clustering with genetic algorithms is used for this purpose as well as in [8], where initial clustering (DBSCAN) was applied on demographic attributes.

Hierarchical clustering was also used in recommender systems [3]. Input data was clustered using hierarchical agglomerative approach, then new items were joined to the most similar cluster in the dendrogram.

Due to its time efficiency, clustering is often applied in mobile phone RS. A recommendation system for tourists [2] is an example, in which clusters are built on users who share similar interests. Data were taken from registering forms and partitioned using k-means algorithm.

3 Description of the Algorithm

A recommender system used in the following experiments is based on clustering approach. It is an algorithm presented in [12], which was modified and adapted to content-based recommendations. Modification concerns the procedure of recommendations generation as well as coping with new articles. It consists of the following procedures: preprocessing and clustering, generation of recommendations, adding new articles.

The operations of preprocessing and clustering are very expensive, therefore they are performed periodically (length of the time interval depends on the number of added posts) in an off-line mode. The steps of this stage are following:

1. Preprocessing of data.
2. Transformation of terms to VSM vectors.
3. Data clustering.
4. Creation of a model of clusters.
5. Calculation of a similarity matrix among the articles from the same cluster.

Preprocessing of text is as follows: first, all capital letters are replaced by small ones, next, the words without any meaning, such as "is" or "the", are removed, and then the remaining text is stemmed. Stemming is a procedure of extracting constant part of words having different form of inflectional. To give an example, "computer", "computers" or "computerization" could have one common stem - "compute". Words after stemming are determined as terms. The benefits from preprocessing are reduction of the number of words and improvement similarity among elements in final clusters.

Before documents are clustered, they are transformed from their letter representation to numbers in Vector Space Model (VSM) [11]. The numbers relate to the relevance selected words in particular documents.

The equation describing a document in Vector Space Model is as follows:

$$D_i = (d_{i1}, d_{i2}, ..., d_{in}) \tag{1}$$

where components d_{ij} are positive real numbers referring to the level of description as well as diversification of the individual terms $t_1, \ldots, t_n$ and n is a number of terms selected for representation of documents.

It has been proposed many methods of document description in VSM. One of them is binary representation: if a term from VSM vector is present in the examining document, the relevant component d_{ij} is equal 1. In the other case it is equal 0. More useful are methods based on term or document frequency (TF, DF, TFIDF). In the experiments presented in this paper, the coefficient TFIDF was used, which is described as follows:

$$TFIDF(D_i) = TF(t_j, D_i) \cdot IDF(t_j) \tag{2}$$

where component $TF(t_j, D_i)$ refers to the number of occurrences of term t_j in document D_i (see (3)) and $IDF(t_j)$ (Inverse Document Frequency) refers to the number of occurrences of this term in all documents (see (4)).

$$TF(t_j, D_i) = \begin{cases} 0, & for\ n_{ji} = 0; \\ \frac{n_{ji}}{n_{max},} & for\ n_{ji} > 0. \end{cases} \tag{3}$$

where n_{ji} denotes a number occurrences of term t_j in document D_i and n_{max} denotes maximal number from occurrences of every term from VSM vector.

$$IDF(t_j) = \log\left(\frac{N}{N_j}\right) \tag{4}$$

where N denotes a number of all documents and N_j - a number of documents containing term t_j.

Clusters are created from vectors of articles in VSM space. Every group is then described by its model. Depending on the clustering algorithm the model can be composed of cluster centroids, boundary or representative points. Each object from the model contains the most characteristic words (with the highest attribute's value) in the group which it belongs to.

After clustering a similarity matrix among articles for every group is calculated. As similarity coefficients can be used distance based measures, correlation or cosine values.

The procedure of propositions generation for an active user is based on the clusters model:

1. Identification of the cluster, which the current article belongs to.
2. Finding in the cluster the most similar articles to the active user's visit history.
3. Proposition of the most similar articles sorted in descending order in terms of their similarity values.

The current article of an active user is searched in the created clusters. Then, basing on the similarity matrix of this cluster, the most similar articles are identified and included in a proposition list R in descending order. The summary similarity to an active user's history is taken into consideration.

The procedure of recommendation generation for an active user is presented in Figure 1. It contains matrices of document similarity (M_{sim}) for clusters C_1 and C_2. The cluster C_1 is composed of the following articles: a_1, a_3, a_5 and a_7, whereas the cluster C_2 contains the articles: a_2, a_4 and a_6. The assumed size of the list R is equal 2. In the first step the active user visited the article a_5, which belongs to the first cluster C_1. The most similar articles to a_5 are a_3 and a_7 ($sim(a_5, a_3) = 0.86$ and $sim(a_5, a_7) = 0.77$). In the second step, the active users visited a_3 and the recommendations list was updated to a_7, a_1 ($sim(a_3{+}a_5, a_7) = (sim(a_3, a_7) + sim(a_5, a_7))/2 = 0.825$ and $sim(a_3 + a_5, a_1) = (sim(a_3, a_1) + sim(a_5, a_1))/2 = 0.815$). Finally, after visiting a_6, $R_{\text{active}} = \{a_1, a_4\}$.

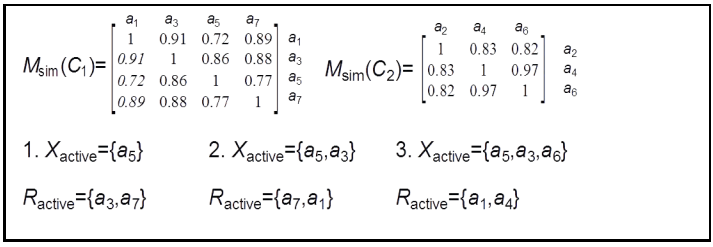


Fig. 1. Example of recommendations generation based on clusters model

An Internet forum is a place, where new articles appear often. To avoid unnecessary re-clustering and make them recommendable a new approach was proposed:

1. Transformation of a new article to a vector in VSM.
2. Classification of the vector to the most similar cluster basing on similarity to the model of groups.

When the time of periodical preprocessing and clustering approaches, new articles are included in input data.

4 Experiments

The recommending algorithm described before was deployed on Internet forum concerning computer science issues. Clustering methods as well as similarity measures were taken from Apache Mahout library (http://mahout.apache.org). To cope with large amount of data and generate recommendations in reasonable time the system was implemented on hdfs file system using Apache Hadoop platform (http://hadoop.apache.org). The first part of the experiments was to

select appropriate clustering method. Effectiveness of grouping results was evaluated with regard to homogeneity of the clusters. Determining the category of every articles in advance the final effectiveness was calculated according to (5), where $Eff(C_i)$ is effectiveness (homogeneity) of cluster C_i, n_{iCat} is a number of articles from the predominant category in cluster C_i and n_{C_i} is a size of this cluster.

$$Eff(C) = \sum_{i=1}^{nc} Eff(C_i), \quad \text{where} \quad Eff(C_i) = \frac{n_{iCat}}{n_{C_i}} \tag{5}$$

The results of the tests on the clustering methods from Apache Mahout library are presented in Table 1. All algorithms were executed with the number of clusters equal 22 (the number of article categories), however the Dirichlet clustering method identified 4 empty clusters. In case of this technique there was also a problem with evaluation homogeneity of one cluster due to its mixture content. The methods: k-means and fuzzy k-means generated comparable results. In case of the fuzzy algorithm the groups had more even sizes, however the original k-means partitioning was a little more accurate.

Table 1. Effectiveness of forum articles clustering

Algorithm	Number of groups	Min group size	Max group size	Min Eff(C_i)	Max Eff(C_i)	Eff(C)
k-means	22	1	26	0.57	1	0.85
fuzzy k-means	22	2	14	0.4	1	0.83
Dirichlet clustering	18	1	31	-	1	-

Effectiveness of the results with regard to similarity measure is presented in Table 2. The mark "-" denotes, that there was a problem with evaluation homogeneity of one cluster due to its mixture content. Finally, k-means clustering method was selected with cosine similarity coefficient.

Table 2. Effectiveness of clustering with regard to similarity measure for k-means method

Similarity measure	Number of groups	Min group size	Max group size	Min Eff(C_i)	Max Eff(C_i)	Eff(C)
Cosine	22	1	26	0.57	1	0.85
Euclidean distance based	22	2	21	0.71	1	0.81
Manhattan distance based	22	1	111	-	1	-

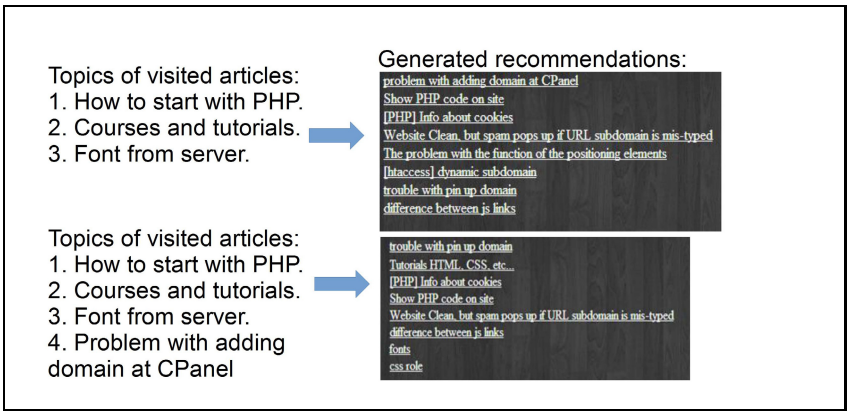


Fig. 2. Example of generated recommendations for the list of visited forum articles

The second part of the experiments was to test the recommender system on clustered data. An example content of a recommendations list is presented in Figure 2. At first, an active user visited 3 topics (on the left) and a list of relevant articles appeared (on the right). Then, the user selected the first of recommended post and the recommendations were updated.

The most common approach to evaluate quality of recommendations of recommender systems is to measure their accuracy. It involves splitting input data into training and testing sets and measuring their RMSE or MAE error for the testing data [10]. However, there are approaches e.g. [7], which argue that the above procedure does not evaluate generated recommendations, but judge the accuracy of individual item predictions. As an example, in travelling recommendations a user have in propositions the places their have already visited.

Due to fact, that the system was deployed in real Internet forum environment, the generated recommendations were evaluated by users. There were selected 15 users, who had the task to visit and assess at least 20 articles. The users rated 763 articles giving 567 (74%) positive notes. A part (7%) of the negatively evaluated articles had high similarity to the topics visited by the users.

Conclusions

The aim of this article was to present a content based algorithm for recommendations generation. The recommendations were created basing on content similarity between articles from on-line forum concerning computer science issues. The similarity calculations were limited to neighbourhood determined by clusters identified by a grouping algorithm. As a clustering method k-means was selected, because its partitioning was the most relevant and homogeneous to categories of articles topics. The recommendations were composed of objects similar to all the articles from an active user's session.

The language of the tested forum was English, however the method can work on text in other languages. It requires replacement of the procedures of text prepro-

cessing into VSM vectors on ones specific to the forum language. Quality of recommendations can be different (rather lower than higher) due to lower effectiveness of the other languages preprocessing procedures. Results of clustering of WWW search results in Polish language in comparison with English is described in [13].

The system was tested by forum users, who gave 74 % positive notes. A quite great part of negatively rated articles was highly similar to the user's session topic, that suggests further modifications of the recommendations generation procedure. It is worth testing the approach, in which to the propositions low similar articles are included.

Acknowledgments. This work was supported by Rectors of Technical University of Bialystok Grant No. S/WI/3/13.

References

1. Anand, S.S., Mobasher, B.: Intelligent techniques for web personalization. In: Mobasher, B., Anand, S.S. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 1–36. Springer, Heidelberg (2005)
2. Gavalas, D., Kenteris, M.: A web-based pervasive recommendation system for mobile tourist guides. Personal and Ubiquitous Computing 15, 759–770 (2011)
3. Haruechaiyasak, C., et al.: A dynamic framework for maintaining customer profiles in e-commerce recommender systems. In: IEEE International Conference on e-Technology, e-Commerce and e-Service, pp. 768–771 (2005)
4. Jain, A.K., Murty, M., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
5. Kim, T.-H., Yang, S.-B.: An Effective Recommendation Algorithm for Clustering-Based Recommender Systems. In: Zhang, S., Jarvis, R.A. (eds.) AI 2005. LNCS (LNAI), vol. 3809, pp. 1150–1153. Springer, Heidelberg (2005)
6. Kużelewska, U.: Advantages of Information Granulation in Clustering Algorithms. In: Filipe, J., Fred, A. (eds.) ICAART 2011. CCIS, vol. 271, pp. 131–145. Springer, Heidelberg (2013)
7. McNee, S.M., Riedl, J., Konstan, J.A.: Accurate is not always good: How Accuracy Metrics have hurt Recommender Systems. In: Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems, pp. 1097–1101. ACM (2006)
8. Moghaddam, S.G., Selamat, A.: A scalable collaborative recommender algorithm based on user density-based clustering. In: 3rd International Conference on Data Mining and Intelligent Information Technology Applications, pp. 246–249 (2011)
9. Pitsilis, G., Zhang, X., Wang, W.: Clustering Recommenders in Collaborative Filtering Using Explicit Trust Information. In: Wakeman, I., Gudes, E., Jensen, C.D., Crampton, J. (eds.) Trust Management V. IFIP AICT, vol. 358, pp. 82–97. Springer, Heidelberg (2011)
10. Ricci, F., et al.: Recommender Systems Handbook. Springer (2010)
11. Salton, G.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18(11), 613–620 (1975)
12. Sarwar, B.: Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. In: 5th International Conference on Computer and Information Technology (2002)
13. Weiss, D.: A Clustering Interface for Web Search Results in Polish and English. Master Thesis, Poznan University of Technology (2001)

Simple Measure of Network Reliability Using the Variance of the Degree Distribution

Ho Tat Lam and Kwok Yip Szeto*

Department of Physics
Hong Kong University of Science and Technology, Clear Water Bay,
Hong Kong,HKSAR, China,
phszeto@ust.hk

Abstract. simple measure of the reliability of a non-regular, undirected, unweighted, connected network is introduced using the variance of the degree distribution of the network. A physical argument for the importance of variance in network reliability, followed by an analytical derivation of the difference between the reliability of a general network and a regular network are presented. The theoretical results are verified by numerical calculation that compares the reliability of networks, taken from an ensemble of networks with same number of nodes and links. The numerical results confirm the negative and linear correlation between the variance of the degree distribution of the network and the logarithm of its reliability. This theory provides a simple and efficient way in the comparison of reliability of two networks with same number of nodes and links.

Keywords: Network reliability, Variance, Degree distribution, All terminal problem.

1 Introduction

Reliability is one of the important measures of how well the system meets its design objective, and mathematically is the probability that a system will perform satisfactorily for at least a given period of time. For a system with many components, the reliability of the system depends on the reliabilities of its components as well as the ways the components are connected. One can increase the system reliability by several means, such as increasing the reliability of the components using parallel redundancy for the less reliable components. Implementation of these steps to improve system reliability will normally consume resource. Thus, a balance between system reliability and resource consumption is essential. In this context, the economics requires a minimization on the cost of the design subject to multi-objective constraints. Beside the reliability of the individual components, the design of network is relevant in many real world applications such as telecommunications [1-3], computer networking [4-6], sewage systems [7], and oil and gas lines [7]. However,

* Corresponding author.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_28,

the network design problem is an NP-hard combinatorial optimization problem [8]. The number of possible network architectures for a given number of nodes (*N*) and links (*L*) will grow exponentially with the size of the network. Moreover, the comparison of the reliability of simple graph, when it is not the usual series or parallel configuration, is itself a difficult problem. Previous approaches have been either enumerative based, that are applicable only for small network sizes [2,6,9], or heuristic-based that can be applied to larger networks but do not guarantee optimality. Examples are found in tabu search [10,11], simulated annealing [1,3,5], and genetic algorithms [7,12-17] for the optimal design of network structure by searching on a large space of possible graphs [18].

In this paper, we focus on the comparison of the reliability of two connected networks with N vertices and L links. This problem can be addressed by the theory of domination by Satyanarayana and Prabhakar [19] and the theory of signature by Samaniego [16,20]. However, as the number of coherent systems of order *N* grows rapidly with *N* [21], general application of these existing theories requires numerical analysis of graphs. For our present work, we consider the all-terminal problem, which addresses the chances of all 'terminals' in a given network being capable to communicate with each other. This network reliability problem relates to the state of the entire system, so that the system works or fails as a function of the working or failure of its components. First let's define the system state with *N* components by the state vector $\vec{x}$ where for each i, $x_i = 1$ if the *i*-th component works and $x_i = 0$ if it fails, so that the space $\{0,1\}^N$ of all possible state vectors for an *N*-component system contains 2^N state vectors. We are interested in knowing if the system as a whole works when the components are in a specific state $\vec{x}$. The mathematical analysis of connectivity can begin with the system *structure function,* $\varphi(\vec{x})$, which is 1 for those state vectors $\vec{x}$ when the system works and is 0 for those state vectors $\vec{x}$ when the system fails. For example, the structure function $\varphi: \{0, 1\}^N \to \{0, 1\}$ for an n-component series and parallel system are

$$\varphi(\vec{x}) = \prod_{i=1}^{N} x_i \quad \text{(series system)}$$

$$\varphi(\vec{x}) = 1 - \prod_{i=1}^{N} (1 - x_i) \quad \text{(parallel system)} \tag{1}$$

In the series case, the system will fail when any component fails: $\varphi(\vec{x}) = 0$ for any $x_i = 0$. In the parallel case, the system fails only when all components fail: $\varphi(\vec{x}) = 0$, only if all $x_i = 0$. For a system of N components at time t, let's denote the probability that the *i*-th component works at time *t* by $p_i = P(X_i = 1)$, where X_i represents the random state of the *i*-th component at time *t*. We define the reliability of a system at time *t* as the probability $h(\mathbf{p})$ that the entire system is

working at that time. This probability $h(\mathbf{p})$ can be computed from the structure function as

$$h(\mathbf{p}) = P(\phi(\mathbf{X}) = 1) = E\phi(\mathbf{X}) \quad \text{with} \quad \mathbf{p} \equiv (p_1, \cdots p_i, \cdots p_N), \tag{2}$$

and $h(\mathbf{p})$is linear in every p_i. The theory of signature [16] handles the calculation of $h(\mathbf{p})$ using the signature vector $\boldsymbol{s}$ which is related to the order statistics of the N component failure times. The application of this theory involves complicated calculation of the minimal path sets of graph.

From this definition of the reliability problem, we see that it is rather difficult to use the theory of signature to compare the reliability of two connected complex networks with same N and L. Here, we introduce a simple method of comparison of networks which is approximate, but efficient. We will formulate a general measurement for reliability of a non-regular complex network based on the property of its degree distribution, specifically on the variance of the distribution. Our method provides a rough guide to reliability, which can then be fine-tuned using the theory of signature [16]. In section 2, we discuss the importance of the degree distribution on reliability, and derive analytically a relation between the reliability of a regular network of given N and L, and the reliability of a complex connected network with the same N and L. In section 3, we substantiate the analytical results with numerical calculation of reliability of a general complex networks with different variance. Finally, we discuss the importance of the simple measure of reliability using variance in section 4.

2 Degree Distribution

The networks with same number of nodes and links have same average degree. Their degree distributions describe the proportions of nodes with different degrees and the degree distribution is an important factor that determines the reliability of networks, even though there are many different topologies corresponding to the same degree distribution. In this section, we first introduce a physical argument for the connection between reliability and the variance of the degree distribution of the network through the concept of isolation. We then present an analytical derivation for the relation between the reliability of a non-regular complex network with a regular network. The relation can be simplified with some approximation, so that the variance of the non-regular network provides a simple measure of the network reliability.

2.1 Isolation and Network Reliability

Certain level of links failure may lead to the complete failure of the network. When the network is divided into two disconnected graph after a link failure, we say that the link failure leads to *isolation*. An isolation which requires least failure of links is the easiest among all isolations and it significantly determines the reliability of the network. For instance, a network with two communities connected by a bridge of only

one link is extremely unreliable because the failure of the bridge immediately leads to the failure of the network. The simplest and most common bridge is the link connecting an individual node to the rest of the network. Thus, we focus on the isolation of a single node.

In the context of all-terminal problem, single node isolation is in general the easiest way to disconnect a network. A small degree node is likely to be disconnected because few link connecting the node to the rest of the network. In comparing two networks with the same number of nodes and links, the average degrees of the networks are thus the same. In a network with larger variance of degree distribution, we have more nodes with small degree and these small degree nodes are vulnerable for isolation. Therefore, we expect that a network with larger variance is less reliable than a similar network with smaller variance, but with the same average degree.

2.2 Analytic Estimation of the Bound on Reliability

For quantitative analysis, we provide an upper bound of the reliability of a connected network whose links fail with probability q=1-p. This estimation provides mathematical substantiation of the heuristic explanation for the negative correlation between variance and reliability.

Suppose the network has N nodes and L links, and the i^{th} node has degree of d_i. In order to maintain the connectedness of the network, every node need at least one working link connecting to the rest of the network and the probability of a node of degree d to have at least one working links is 1-q^d. Then the actual reliability of the network is bounded above by the following estimate R_0,

$$R \le R_0 = \prod_{i=0}^{N} (1-q^{d_i}) = \prod_{d=1}^{\infty} (1-q^d)^{N(d)} \tag{3}$$

where $N(d)$ is the number of nodes of degree d. Taking the logarithm, we obtain

$$\ln R \le \ln R_0 = \sum_{d=0}^{\infty} N(d) \ln(1-q^d) \tag{4}$$

Now, the number of nodes and links are conserved which give us two constraints,

$$N = N(1) + N(2) + \sum_{d=3}^{\infty} N(d) \tag{5}$$

$$2L = N(1) + 2N(2) + \sum_{d=3}^{\infty} dN(d) \tag{6}$$

With the two constraints, we can solve for $N(1)$ and $N(2)$ in terms of N, L and the summation term starting with d=3. We can thus rewrite $\ln R_0$, without $N(1)$ and $N(2)$, as

$$\ln R_0 = N \ln\left(\frac{1-q}{1+q}\right) + 2L\ln(1+q) + \sum_{d=3}^{\infty} N(d) \ln\left[\frac{1-q^d}{(1-q)(1+q)^{d-1}}\right] \quad (7)$$

Here, this formula shows that R_0 is positively correlated with L and negatively correlated with N. In another word, a network with given N, the more links we provide to the network, the higher its reliability. It follows our intuition on reliability.

Assuming $N(d)$ is a continuous function of d, we first construct a regular network where every nodes have same degree $d^* = 2L/N$. The degree distribution for this regular network is $N(d^*) = N$ and $N(d) = 0$ for other d. For such regular network, its estimated reliability is R_0^*, and its logarithm is

$$\ln R_0^* = N \ln(1-q^{2L/N}) \quad (8)$$

We now look at the difference of the logarithm of the estimated reliability of a normal non-regular network and this regular network. One can show that there exists d' in the interval $d' \in (d, d^* = 2L/N)$ such that

$$\ln R_0 - \ln R_0^* = \sum_{d=3}^{N} N(d) \times (d - d^*) \times F(d', q)$$

$$where \qquad F(d', q) = \left.\frac{\partial^2 \ln R_0}{\partial N(d') \partial d'}\right|_{d' \in (d, d^* = 2L/N)} \quad (9)$$

We now repeat the argument and say that there exists $d'' \in (d, d^* = 2L/N)$ such that

$$\ln R_0 - \ln R_0^* = \sum_{d=3}^{N} N(d) \times (d - d^*) \times (d - d^*) \times G(d'', q)$$

$$where \qquad G(d'', q) = \left.\frac{\partial F(d', q)}{\partial d'}\right|_{d'' \in (d, d^*)} \quad (10)$$

After some algebra, one shows that G is always negative, meaning that $R_0 \leq R_0^*$. This analysis thus proves that the regular network, with $d^*=2L/N$ has a higher reliability than a normal non-regular network with the same N and L. This result may appear to be the final answer to the design of the most reliable network: just get a regular network which is most homogeneous and symmetric one can obtain for given N and L. However, unfortunately, a regular network for general N and L usually cannot be constructed as $d^*=2L/N$ is generally not an integer.

Now, we can go a bit further with this analysis by the following approximation

$$\begin{aligned} \ln R_0 - \ln R_0^* &= \sum_{d=3}^{N} N(d) \times (d - d^*)^2 \times G(d'', q)\Big|_{d'' \in (d, d^*)} \\ &\approx \frac{\partial^2 \ln R_0}{\partial N(d) \partial^2 d}\Bigg|_{d=d^*} \sum_{d=3}^{N} N(d) \times (d - d^*)^2 \\ &= -VAR \times N \times \frac{(\ln q)^2 q^{d^*}}{(1 - q^{d^*})^2} \end{aligned} \tag{11}$$

where *VAR* is the variance of degree distribution. The approximation is valid when the variance is small. We can see that regular network is most reliable among all networks with same number of nodes and links. Furthermore, network with a broader degree distribution has a larger variance, leading to a lower reliability. The logarithm of network reliability is approximately linear in the variance of the degree distribution.

3 Numerical Test

For numerical calculation of reliability, we use Monte Carlo simulation to compute the probability of system working as a function of the robustness of the link. We first generate an ensemble of connected networks with the same connectivity (same *N* and *L*). For each network in the ensemble, we randomly cut links with probability q=1-p, so that the resultant networks can be either connected or disconnected. We can then get a numerical value of the reliability of this topology by computing the ratio of number of networks that is connected in the ensemble over the total number of networks in the ensemble.

In this section, we numerically construct random networks and use them to verify the prediction in Section (2). We restrict ourselves to networks which have same number of nodes and links. We first generate 1000 networks and compute their reliability and variance of degree distribution. The networks with same variance are grouped together and we take the average and the standard deviation of the logarithm of their reliability. This allows us to obtain a relation between logarithm of reliability and variance with the standard deviation as their error bar. Averaging reliability is essential. Two different networks may have same variance but different reliability. Averaging reliability among the group of networks with same N and L and variance in degree distribution reduces the random error of numerical computation on reliability and also averages the difference between different topologies with same variance. We illustrate in Fig.1 the numerical result of the logarithm of reliability for networks with same number of nodes and links, but different variance. We observe in Fig.1 that the logarithm of reliability is negatively and linearly correlated with variance. It supports the theoretical analysis in Section (2). We define the slope of the

linear regression to be the reliability decay rate. In Fig.2, we show the dependence of reliability decay rate on the number of nodes for networks with average degree of 3. The relation is linear as predicted by theoretical analysis.

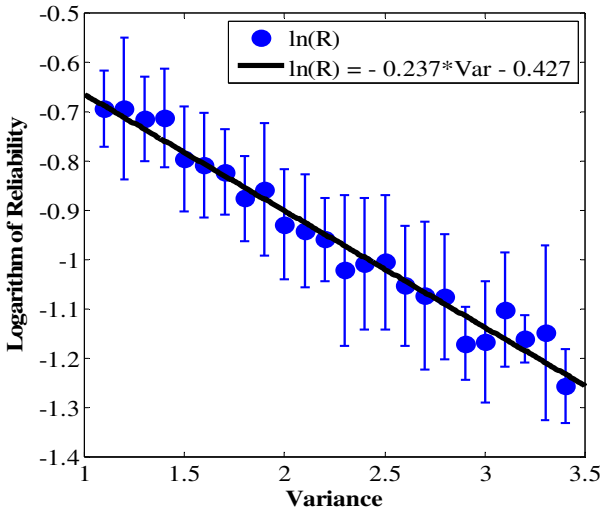


Fig. 1. Logarithm of reliability versus the variance for networks with 20 nodes and 30 links

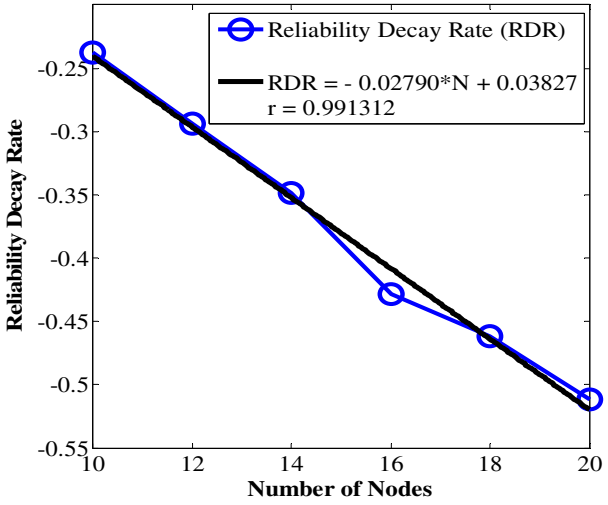


Fig. 2. Reliability decay rate versus the number of nodes for networks with average degree of 3.The robustness of each link is set to be $p = 0.7$.

From our numerical results we can deduce a relation between the reliability and the variance of the degree distribution of the network of N nodes and average degree d,

$$R \approx R' e^{-VAR \times N \times F(d)} \tag{12}$$

where R' is a constant and $F(d)$ is a monotonic increasing function of the average degree. We see that reliability decays exponentially with the variance of degree distribution and the number of nodes. Our numerical results show that the formula is valid for a wide range of networks.

4 Discussion

In conclusion, our analysis on the relation between reliability and variance provides an alternative way to understand network reliability. We have given a physical explanation of the correlations and substantiate our theory with mathematical analysis and numerical result. Our theory also suggests a direction for the design of the most reliable network with same number of nodes and links, by approaching a regular network whose nodes all having the same degree.

Our theory has its own limitation. Due to the approximation applied in the theory, the mathematical relation we obtained is valid for networks with high reliability. As we pointed out earlier, two networks with same *N, L*, and degree distribution can still have different reliability, dependent on the exact topologies of the networks. Our present simple measure using the variance in degree distribution cannot distinguish the reliability of two networks with same *N, L* and same variance of degree distribution. Thus, in the ranking of the reliability for networks, a more detailed analysis on the relation between topology and reliability of networks will be required. Indeed, our theory is incomplete as we only consider some of the simplest properties of the network associated with its degree distribution. Furthermore, we know that there exist many networks with the same degree distribution and we need a more sophisticated theory to distinguish the difference in their reliability, not simply by going to higher moments of the distribution. We expect that a more detailed relation between the topology and the network reliability exist that provides finer classification of networks with same *N, L* and degree distribution. This observation shows that our simple measure of variance in the comparison between two networks with same average degree is only a crude measure. However, the computation of variance is simple for large complex networks and our analysis show that using variance is an efficient first step in the analysis of reliability. Indeed, we propose that we should first compute the variance of the network before we go deeper into the comparison of the reliability of networks, either numerically or with more powerful mathematical analysis.

Acknowledgement. K.Y. Szeto acknowledges the support of grant FS-GRF13SC25 and and FS-GRF14SC28.

References

1. Aiiqullah, M.M., Rao, S.S.: Reliability optimization of communication networks using simulated annealing. Microelectronics and Reliability 33, 1303–1319 (1993)
2. Jan, R.-H., Hwang, F.-J., Chen, S.-T.: Topological optimization of a communication network subject to a reliability constraint. IEEE Transactions on Reliability 42, 63–70 (1993)
3. Pierre, S., Hyppolite, M.-A., Bourjolly, J.-M., Dioume, O.: Topological design of computer communication networks using simulated annealing. Engineering Applications of Artificial Intelligence 8, 61–69 (1995)
4. Aggarwal, K.K., Chopra, Y.C., Bajwa, J.S.: Topological layout of links for optimizing the overall reliability in a computer communication system. Microelectronics and Reliability 22, 347–351 (1982)
5. Fetterolf, P.C., Anandalingam, G.: Optimaldesign of LAN-WAN internet works: an approach using simulated annealing. Annals of Operations Research 36, 275–298 (1992)
6. Wilkov, R.S.: Design of computer networks based on a newreliability measure. In: Fox, I. (ed.) Proceedings of the Symposium on Computer-Communications Networks and Teletraffic, pp. 371–384. PolytechnicInstitute of Brooklyn, Brooklyn (1972)
7. Walters, G.A., Smith, D.K.: Evolutionary design algorithmfor optimal layout of tree networks. Engineering Optimization 24, 261–281 (1995)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Co., San Francisco (1979)
9. Aggamal, K.K., Rai, S.: Reliability evaluation in computercommunication networks. IEEE Transactions on Reliability R-30, 32–35 (1981)
10. Glover, F., Lee, M., Ryan, J.: Least-cost network topology design for a new service: an application of a tabu search. Annals of Operations Research 33, 351–362 (1991)
11. Koh, S.J., Lee, C.Y.: A tabu search for the survivable fiber optic communication network design. Computers and Industrial Engineering 28, 689–700 (1995)
12. Dengiz, B., Altiparmak, F., Smith, A.E.: Efficient optimization of all-terminal reliable networks using an evolutionary approach. IEEE Transactions on Reliability 46, 18–26 (1997)
13. Kumar, A., Pthak, R.M., Gupta, Y.P., Parsaei, H.R.: Agenetic algorithm for distributed system topology design. Computers and Industrial Engineering 28, 659–670 (1995)
14. Kumnar, A., Pthak, R.M., Gupta, Y.P.: Genetic-algorithm-based reliability optimization for computer network expansion. IEEE Transactions on Reliability 44, 63–72 (1995)
15. Deeter, D.L., Smith, A.E.: Heuristic optimization ofnetwork design considering all terminal reliability. In: Proceedings of the Reliability and Maintainability Symposium, pp. 194–199. IEEE, Piscataway (1997)
16. Samaniego, F.J.: System signatures and their applications in engineering reliability. Springer Science+Business Media, LLC, Boston (2007)
17. Kuo, W., Prasad, V.R., Tillman, F.A., Hwang, C.: Optimal Reliability Design. Cambridge University Press, Cambridge (2001)
18. Colbourn, C.J.: The Combinatorics of Network Reliability. Oxford University Press, Oxford (1987)

19. Satyarananaya, A., Prabhakar, A.: A New Topological Formula and Rapid Algorithm for Reliability Analysis of Complex Networks. IEEE Transactions on Reliability TR-30, 82–100 (1978)
20. Boland, P.J., Samaniego, F.J.: The Signature of a Coherent System and its Applications in Reliability. In: Soyer, R., Mazzuchi, T., Singpurwalla, N. (eds.) Mathematical Reliability: An Expository Perspective, pp. 1–29. Kluwer Academic Publishers, Boston (2004)
21. Kleitman, D., Markowsky, G.: On Dedekind's problem: the number of isotone Boolean functions. II. Transactions of the American Mathematical Society, A 213, 373–390 (1975)

CDM: A Prototype Implementation of the Data Mining JDM Standard

Piotr Lasek

Chair of Computer Science, University of Rzeszów
ul. Prof. St. Pigonia 1, 35-310 Rzeszów, Poland
lasek@ur.edu.pl

Abstract. There exists a great variety of tools and applications designed for data mining and knowledge discovery. Historically, from the 1970s, a number of available tools continues to grow. For this reason, a potential user may have difficulties when trying to choose an appropriate tool for himself. Similarly, when it comes to the implementation and evaluation of newly proposed data mining algorithm, an author needs to consider how to verify his proposal. Usually, a new algorithm or a method is implemented and tested without using any standardized software library and tested by means of an ad hoc created software. This causes difficulties in case when there is a need to compare efficiency of two methods implemented using different techniques. The aim of the paper is to present a prototype implementation of a data mining system (CDM) based on the Java Data Mining standard (JDM) that provides standardized methods designed for convenient implementation and verification of data mining algorithms.

Keywords: Data mining, clustering, JDM standard, CDM.

1 Introduction

Data mining is a relatively new and rapidly developing field of computer science. Its main goal is to explore data so that new, unknown and potentially useful patterns could be discovered. Data mining methods employ different and specialized algorithms for building, modeling and evaluation of discovered knowledge and are used to analyze multiple kinds of data from many domains such as medicine, healthcare, finance, telecommunication, science, etc. Recently, data mining tools, crucially improved and simplified data mining by employing new efficient algorithms [3, 4, 6]. The number of methods and tools both commercial and open source increases rapidly every year [2, 5, 7]. Undoubtedly, data mining tools and applications become widely known and used when it comes to exploring knowledge from large amounts of data.

Data mining techniques evolve so that they become both more expert and problem oriented [14]. Additionally, during recent years, they also became more useful in everyday applications, such as, for example, e-mail filtering or credit card fraud analysis [15]. For professionals, the data mining process is more like an art because of the fact that they usually limit their analysis to several best known techniques. On the other

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_29,

hand, beginners are often overwhelmed by the variety of methods and the diversity of existing different, both commercial and open source, data mining tools. The diversity of these tools was a subject of numerous works [2]. The aim of our work is not to compete with existing software but rather to present and promote the way how to implement and test data mining algorithms so that they could be easily reused by another users.

Among numerous tools and data mining libraries the introduction of the Java Data Mining standard (JDM) is a step towards standardization and vendor neutral development of data mining solutions. JDM was designed so that it is based on solid concepts such as so-called mining objects, models and tasks. On the other hand, despite it comes with standard, its Application Programming Interface (API) is also flexible and extensible.

The paper is divided into four sections. After the introduction we recall basic definitions of terms used in the next part of the paper which are related to the implementation of the prototype CDM (for *Common Data Mining*) system based on the JDM standard. Additionally, in the second section we briefly describe the JDM standard and present, from our perspective, its crucial features. In Section 3 we describe our prototype implementation of data mining engine and several other components required to meet the standard of JDM. We summarize our paper as well as present our further plans related to development of CDM in Section 4.

2 The Architecture of JDM

JDM was created by high-class experts and meets the following crucial assumptions: addresses a large developer community, is a standard interface, has a broad acceptance among vendors and consumers, is extensible, simplifies data mining for novices while allowing control for experts, recognize conformance limitations for vendor implementations, supports requirements of real, industrial applications, appeals to vendors and architects in other development domains [3]. For these reasons we have chosen JDM as a base for implementation of our prototype CDM.

The specification of JDM was accepted by Java Community Process Executive Committee in 2004. The architecture of JDM comprises three main components, namely: API, DME (Data Mining Engine) and MR (Meta Data Repository). API (Application Programming Interface) is the set of programming interfaces which should be implemented so as to provide access to services available in DME. Data Mining Engine can be implemented in several ways. One possible way is to implement it as a library providing methods to access data to be analyzed, another may be to implement it as a more convenient tool such as a server. In the latter case, such an implementation of DME is usually called DMS (Data Mining Server). Next, the repository of meta data is used to store so-called Data Mining Objects which can be used in different steps of mining process. Repositories can use flat file system or can be programmed as a relational database similarly to the Oracle Data Mining implementation [1]. Moreover, if necessary, it is possible to add additional components that are not included in the specification of the JDM standard.

In Figure 1 we recalled the diagram of several so-called Mining Objects used in JDM. A Mining Object is a base class for all classes in JDM and comprises the basic and common features such as: name, description, identifier or type of an exploration object. A Mining Object can be saved, under a specific name in a Mining Objects Repository and during a mining process, it can be accessed by another methods by using its name. In JDM the data mining objects are divided into the following types: Data Specification Objects, Settings Objects and Tasks. Data Specification Objects are designed for defining input data by using both logical and physical interfaces. For example the Logical Data based classes are used to interpret data whereas Physical Dataset based classes designed to define the place where data are stored as well as the attribute names and data types. By means of these two types of data classes (logical and physical) it is possible to separate data from algorithms. For example, by using Settings Objects, it is possible to provide parameters to data mining functions. There are different kinds of Setting Objects classes which are appropriate for different kinds of mining functions, namely: Clustering Settings, Supervised Settings, Attribute Importance Settings, Association Settings. Settings Objects provide means to control the build and apply process by setting values of processes or algorithms parameters. For example Build Settings based classes are used to specify settings at the function level and optionally at the algorithm level. Apply Settings classes for instance, provide flexibility when defining the results from model apply.

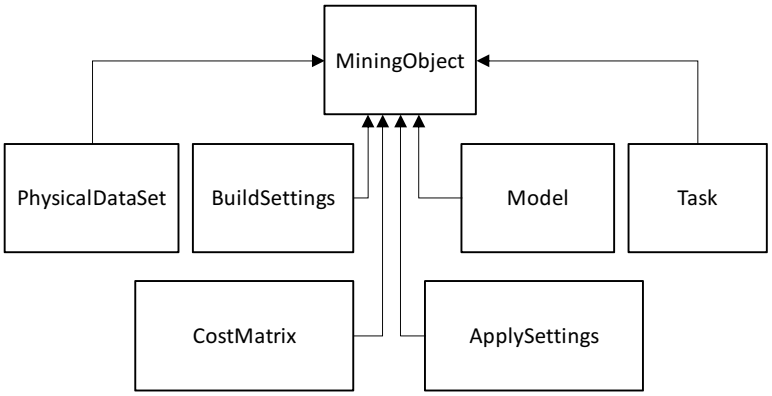


Fig. 1. A simplified class diagram of named object of JDM

Model Objects are used to store a compact representation of the knowledge. This kind of objects provides details at the function and algorithm level. The *Model* interface is the base interface for all models used in JDM. It comprises another interface called *ModelDetail* encapsulating algorithm-specific details. A sample implementation of the model for classification could consist of the following classes *ClassificationModel*, *ClassificationSettings* and *TreeModelDetail*. The first class would provide common content for all kinds of classification algorithms and the *TreeModelDetail* class would provide elements specific to the algorithms based on the decision tree.

Another important element of the JDM architecture is the entity called Task. Tasks represent all information that is required to perform mining operation. Tasks are

started by invoking the *execute* method from the *Connection* object. Because of the fact that when analyzing big data sources mining tasks can be long running, the JDM architecture supports both synchronous and asynchronous execution of tasks. The tasks can be controlled by the handle represented by an object of the *ExecutionHandle* class.

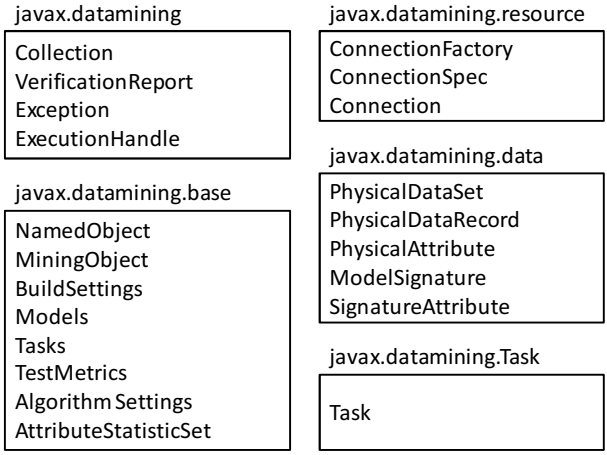


Fig. 2. The set of interfaces (by packages) to be implemented to meet the JDM standard

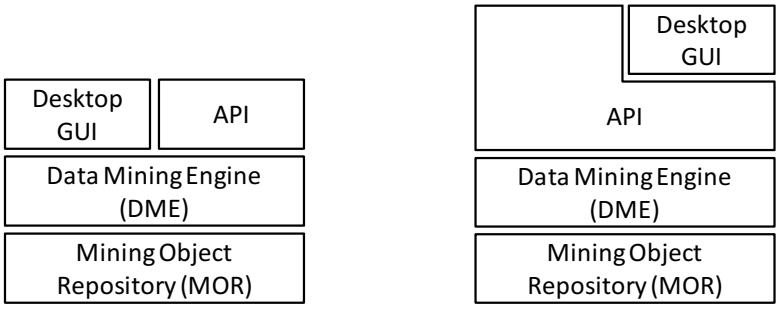


Fig. 3. A sample data mining tool's architecture possible to be implemented using JDM

The JDM architecture allows development of customized implementations of standard interfaces. In order to create an individual implementation based on the JDM standard, the minimum set of interfaces must be implemented. The minimum implementation comprises elements presented in Figure 1. The crucial elements are listed below:

- *Connection*, *ConnectionFactory*, *ConnectionSpec* – these classes are designed to provide access to data and dispatch execution of mining functions according to the specification of the mining experiment.
- *PhysicalDataSet*, *PhysicalDataRecord*, *PhysicalAttribute* – these classes are used to represent datasets as well as data records and attributes. Additionally factories for creation of object of these classes should be implemented. Optionally, if necessary *LogicalData* and *LogicalAttribute* can be implemented.
- Basic implementation of the *Task* interface capable of, at least, synchronous execution of a mining operation with an accompanying factory class to create objects of concrete tasks classes.

In Figure 3 we have recalled two possible architectures of an implementation of the JDM standard to show that the standard is flexible and extensible. For example, when the system is created following the architecture presented in Figure 3a, a user is able access the data mining engine (DME) directly (via graphical user interface) or by means of API. On the other hand, as presented in Figure 3b, the system can be programmed so that the data mining engine cannot be accessed directly, but by public API. Vendors implementing their own JDM based systems can select the most appropriate approach from their perspective. For example, the system can read data from files and, on the other hand, can be more database-centric by reading and storing data in a relational database.

The goal of our work was to create a simple tool for performing ad hoc data mining task by providing functions for loading data from files containing sample benchmark data. We decided to use JDM as a standard framework in order to ensure that created methods could be easily executed in different environments supporting the JDM standards. This could give the possibility to implement and test the implemented algorithm using one system and deploy it easily in another one.

3 The Implementation

All interfaces in JDM are defined as a pure Java specification. For this reason all classes implementing JDM interfaces can be programmed also purely in Java. Nevertheless, vendors have possibility to implemented their own methods behind the JDM interface so that any implementation or technology can be used. In other words, vendors have possibility to wrap up any kind of a source code with the JDM interfaces.

In this section we present how we created the prototype implementation of the system based on the JDM interfaces. First of all we determined and created the minimal set of classes to be implemented to meet the minimum implementation of a functional JDM system. This task required over a dozen classes to be created. In Figure 4, on the left side, we presented those classes, however, for the sake of simplicity, some of optional classes were not shown in the figure. On the right side, we shown how to derive own customized classes from the minimum implementation classes in order to create an implementation of a new algorithm, for example. In our case, we implemented within CDM several clustering algorithms, such as widely known *k-Means* [8] and two version of the *NBC* density based algorithm [9]. One can easily notice that in our custom implementation of the clustering module, it was necessary to create a

package called *javax.datamining.clustering* containing classes deriving from classes implemented in the *javax.datamining.data* package and implementing appropriate JDM interfaces. The next step was to create another package intended for the custom implementation of a single clustering algorithm. Inside this package we placed only two classes extending two base classes such as: the *CDMBasicClusteringSettings* class and the *CDMBasicClusteringAlgorithm* class.

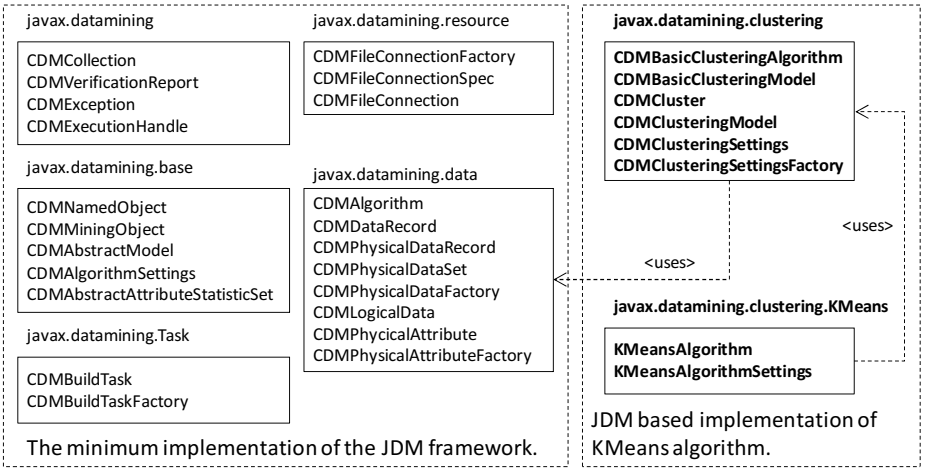


Fig. 4. The prototype implementation (CDM) of the JDM mining system

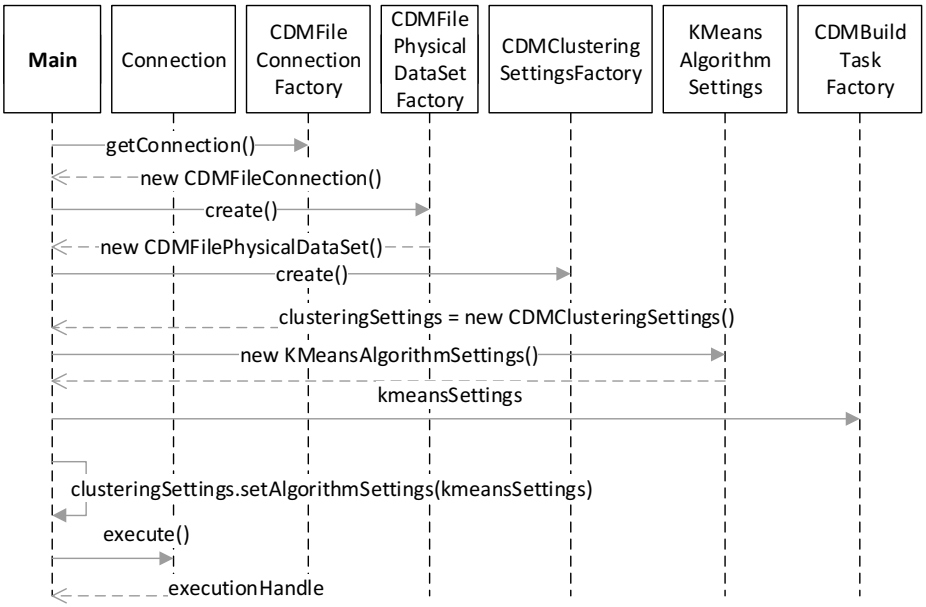


Fig. 5. A simplified sequence diagram presenting the execution of the implemented *k-Means* algorithm

From the perspective of a data mining programmer, when adding a new clustering algorithm to CDM, there are only two classes to be implemented, namely the classes located in a package of a new algorithm. However, if one would like to add an algorithm belonging to the data mining function that is not yet supported by CDM, it would be necessary to create a new package, for example a classification package called *javax.datamining.classification*, and implement the basic classes related to the domain of classification. For example, it might be necessary to create classes such as: *CDMBasicClassificationAlgorithm*, *CDMBasicClassificationModel*, *CDMClass*, *CDMClassificationModel*, etc. – similarly to classes implemented and located in the clustering package.

In Figure 5 we presented the simplified sequence diagram explaining how the sample *k-Means* clustering algorithm is executed. The process of execution of an algorithm comprises several steps. First, it is required to create an object of connection representing a connection to a Data Mining Repository. In our implementation, the connection class was called *CDMFileConnection*. It provides method for reading data from a file system. The path to the source of data (a text file) is given in a form of URI. Then, after creating the connection object using a connection factory, the dataset needs to be prepared. It is done by using *CDMFilePhysicalDataSetFactory* object which creates a *CDMFilePhysicalDataSet* object. Next, co-called physical attributes, are added to the dataset object (this was not shown in the figure) in order to define the structure of the dataset. For example, in this step it is possible to specify which attributes will be taken into account during further experiments and what are the types of those attributes. Attributes, created by means of the *CDMPhysicalAttributeFactory*, are added to the physical dataset. Finally, the defined physical dataset is saved into the Data Mining Object repository by invoking the *saveObject* method which is available in the connection object. The next step is the preparation of the clustering algorithm. Usually every algorithm takes one or more parameters, so at the beginning it will be important to create the settings object (by means of the *CDMClusteringSettingsFactory* class) to make possible to set the appropriate algorithm parameters. The settings factory object returns an object of the *CDMClusteringSettings* class, and then, by using methods provided by the settings object, the user has possibility to set values of parameters of the algorithm. The settings are saved into the data mining repository using the *saveObject* method from the connection object. The last part of the algorithm preparation is the creation of the build task. The build task, represented by an object of the *CDMBuildTask* class, is created by the build task factory (*CDMBuildTaskFactory*). The build task is designed to specify a task that integrates the dataset, the settings (in which the name algorithm used in an experiment is passed) and an output model into which the model of discovered groups will be written. After performing the above actions, namely, specifying the data source, the algorithm and its parameter as well as the output, it is now possible to run the algorithm. It is done by invoking the *execute* method which is provided by the connection object. The execute method returns an execution handle that can be used to control long running task, however, the current implementation of CDM does not provide a possibility to execute long running tasks yet. After the algorithm is ended, it is possible to get results by means of the *retrieveObject* method from the connection object. According to the

specification of JDM, when it comes to clustering, the output of the clustering comprises discovered clusters and rules. So, in the next step it is possible to take advantage of the discovered clusters, for example by verifying or visualizing discovered groups.

4 Conclusions and Further Works

In this paper we have presented the data mining system (CDM) based on the Java Data Mining standard. The interfaces provided by the JDM standard allow creation of the tool supporting all steps of typical process of data mining, such as: data integration, selection, cleaning, integration, data mining, pattern evaluation [13]. CDM covers the minimum implementation required for the system to comply with the JDM standard as well as several clustering algorithms. We consider that using a standardized tool for implementation and testing of new algorithms gives an opportunity to all interested data mining programmers to create software so that it can be easily used and tested in another projects.

The source code of the implemented system is currently available under the following location: http://rspn.univ.rzeszow.pl/?p=682. It can be downloaded using any SVN client. The source code is available in the form of a project of the *Eclipse* platform and it can be run and debugged.

Since our interests are mostly focused on clustering methods, in the nearest future, we will undoubtedly extend our implementation by adding to it more clustering algorithms such as *DBSCAN* [10], *TI-DBSCAN* [11] and *TI-NBC* [12], as well as constrained versions of these algorithms. In cooperation with other authors another missing data mining functions will be gradually added to system.

References

1. Oracle Text. Oracle Text Application Developer's Guide 1Og Release 2
2. Mikut, R., Reischl, M.: Data mining tools. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(5), 431–443 (2011)
3. Hornick, M.F., Marcadé, E., Venkayala, S.: Java data mining: strategy, standard, and practice: a practical guide for architecture, design, and implementation. Morgan Kaufmann (2010)
4. Goebel, M., Gruenwald, L.: A survey of data mining and knowledge discovery software tools. ACM SIGKDD Explorations Newsletter 1(1), 20–33 (1999)
5. Kurgan, L.A., Musilek, P.: A survey of Knowledge Discovery and Data Mining process models. Knowledge Engineering Review 21(1), 1–24 (2006)
6. Mariscal, G., Marbán, Ó., Fernández, C.: A survey of data mining and knowledge discovery process models and methodologies. Knowledge Engineering Review 25(2), 137 (2010)
7. Ruotsalainen, L.: Data mining tools for technology and competitive intelligence. VTT (2008)
8. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) 28(1), 100–108 (1979)

9. Zhou, S., Zhao, Y., Guan, J., Huang, J.: A neighborhood-based clustering algorithm. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 361–371. Springer, Heidelberg (2005)
10. Ester, M., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96 (1996)
11. Kryszkiewicz, M., Lasek, P.: TI-DBSCAN: Clustering with DBSCAN by means of the triangle inequality. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 60–69. Springer, Heidelberg (2010)
12. Kryszkiewicz, M., Lasek, P.: A neighborhood-based clustering by means of the triangle inequality. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 284–291. Springer, Heidelberg (2010)
13. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
14. Liao, S.-H., Chu, P.-H., Hsiao, P.-Y.: Data mining techniques and applications–A decade review from 2000 to 2011. Expert Systems with Applications 39(12), 11303–11311 (2012)
15. Serban, F., et al.: A survey of intelligent assistants for data analysis. ACM Computing Surveys (CSUR) 45(3), 31 (2013)

Confidential Transportation of Data on the Technical State of Facilities

Dariusz Laskowski and Piotr Łubkowski

Institute of Telecommunications, Faculty of Electronics, Military University of Technology
Gen. S. Kaliskiego 2 Street, 00-908 Warsaw
{dlaskowski,plubkowski}@wat.edu.pl

Abstract. Transfer of information, between both the operator and the technical object (machine, robot, network, etc.) is related to the transport of data. This process is realized through terminals and interfaces for fixed and mobile, simple and complex structures of the system (usually the networks, such as telecommunication network, IP network, etc.). The required level of security can be achieved during both development of new and modernization of existing technical objects. Exploitation of technical systems enforces use of confidentiality mechanisms such as those related to the IPSec (Internet Protocol Security), DM/VPN (Dynamic Multipoint/Virtual Private Network), VLAN (Virtual Local Area Network) or FW (Firewall). Therefore, the paper proposes a complementary model of secure data transport which can be applied in a decentralized data base containing information about the state of a technical object. The models of secure data transfer used by IPSec, VPN and VLAN have been tested in several scenarios reflecting the transfer of sensitive data in the industrial sector between the Authorized Service Stations and the Head Office of the company. Performance tests and the obtained results are resultant of repetitive output data obtained on the basis of numerous experiments.

Keywords: Internet Protocol, virtual networks, web filtering, secure data.

1 Introduction

Transfer and/or transformation of information, between both the operator and the technical object (machine, robot, network adapter, etc.) is related to the transport of data through terminals and interfaces for fixed and mobile, simple and complex structures of the system (usually the networks, such as telecommunication network, data communication network - IP network, etc.). Standardization of the points of contact between the objects determines the desired transport of data in relation sender - addresser, forming the basis for the implementation of a specific type of service. Another determinant is the adherence to the criteria adopted in the form of a defined set of properties of the system (e.g. in the form of potentiality, efficiency, quality, reliability, security, etc.) affecting the proper operation of the system and its components in terms of subjectivity or objectivity [1]. Network security has a big impact on of companies functioning (i.e. profit, prestige) and public organizations (i.e. to protect people). In many cases, security products (devices and applications) are not very

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_30,

effective in heterogeneous networks. This is due to the fact, that they are generally available. It is possible to increase their effectiveness by adding their own products. Therefore, the article presents the concept of confidential delivery of services such as transportation data (including voice - VoIP). The proposal for heterogeneous network uses products:

1) Classical: IPSec, virtual networks (VLAN, DM/VPN).
2) Proprietary solutions: **s**tate **f**ull **i**nspection and **p**acket-**f**iltering firewalls (**SFIPF FW**), **S**ecure **S**oftphone application using **PJ**SIP library (**SSPj**).

The combination of commercial products with dedicated applications will be examined in a test environment (test bed).The test bed is the appropriate platform to verify the developed products in a heterogeneous network with IP-compatible Protocol Stack. Functionality of SFIPF FW and Softphone will be tested in many different scenarios research. To test bed will be added the EMC test modules for electromagnetic compatibility testing. The example of the EMC test algorithm was proposed by Nowosielski [2]. The test algorithm was tested during interlaboratory comparison testing described by Nowosielski [3]. Proposed algorithm will be used in future application of network security.

2 The Sensitivity of Data

Therefore, it is reasonable to emphasize that the service delivery environment should be provided with mechanisms for anomalous detection (occurring during the operation - in particular use) and optimization (their destructive impact) submitting data on the basis of which there is a chance to obtain reliable answers to the following questions:

1) Is it possible to detect anomalies as they arise?
2) Is it possible to identify the "infected" elements and the data that have been disclosed?
3) Is it possible to respond with sufficient effectiveness while minimizing the escalation of destruction?
4) Is it possible to estimate the losses in terms of subjectivity and objectivity (prestige, sensitive data)?
5) In what state of suitability a technical object is currently and what tasks are possible to complete to move to the next state?

The main steps of the process of dealing with this issue are presented in the figure (Fig. 1.) depicting cycle of anomalies "life".

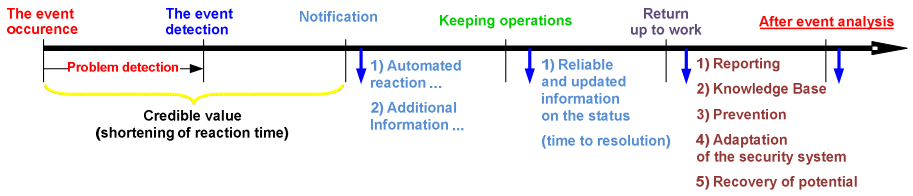


Fig. 1. The cycle of anomaly "life"

Taking into account the possible occurrence of an event consisting in risk of data loss during the delivery of services, the following activities may be performed:

1) Acceptance: to be aware of and assess risk levels for each type of threat.
2) Reduction: have mechanisms reducing the risk and accept the residual risk remaining after the application is deployed.
3) Movement: transpose insurance to contracts with subcontractors, partners.

These tasks aimed at optimizing the level of risk of data loss are used adequately to the level of importance of sensitive data stored in dedicated data stores (storages repositories).The examples of sensitive data used for the identification and assessment of OTS are information transmitted in the industrial sector between Departments and Head Office of Companies (Fig. 2.), i.e.: information on the vehicles or equipment damage occurring during the operation. Protection of information may indicate the authority and prestige of the company as well as the number sold products.

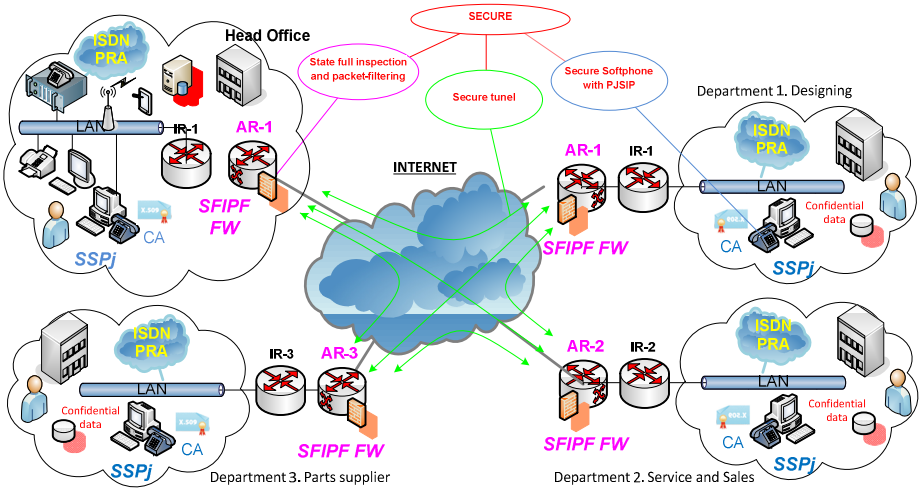


Fig. 2. Proposed use of proprietary solutions for business

3 Service Security

Integrated systems and telecommunication networks constitute a platform for exchange of information between users (network operating terminals located in different dislocations - Departments). If necessary, they can be supplemented with ICT components that can support safety, reliability and quality of data transport in the end-to-end service provision. The effective implementation of data exchange in the structures of the organization of any industry is achieved through the use of computer networks connecting multiple nodes to form a coherent net-work based on the available transport resources (usually the Internet and / or intranet). Network stations are terminals (PCs, smart phones), access points (wired and wireless), hubs (second layer switches), traffic control elements (routers, third layer switches), service servers, links

(twisted-pair, fiber or wireless), printing and graphical devices (i.e. printers) etc. An important aspect is also the flexibility of variable requirement in terms of multiplicity and variability and the reproducibility of the transaction of data flows generated by users with different centers of operation ranging from the radio - (GSM/UMTS/LTE) through wired (LAN/MAN/WAN) up to the fiber environment (FTTx).

Based on the analysis made above concerning the network environments, the storage of data constituting a single structure on multiple servers using a connection to the database system through commercially available resources such as telecommunications links (Public Switched Telephone Network) or the Internet seems to be a reasonable approach. The advantage of this solution in comparison to a central database, or a number of local databases is the reduction of network traffic generated by client terminals, less risk of loss of data and improved reliability and performance of the system also through a central repository. Reliability of database forms a framework of data transmission in the operational decision making process and enforces the need for authorized access to defined areas (raster) for data insertion, their updates and acquirement. Detection and protection systems are containing the event logs on access and modifications to the information stored in the database. Apart from the standard solutions, the recommended communication between the databases must also take into account dedicated solutions, i.e.:

1) **S**tate **f**ull **i**nspection and **p**acket-**f**iltering firewalls (SFIPF FW).
2) **S**ecure **S**oftphone application using **PJ**SIP library (SSPj).
3) Confidentiality of data transport channels VPN built based on protocols such us IPSec, SSL and data encryption e.g. Advanced Encryption Standard (AES) 512.

4 State Full Inspection and Packet-Filtering Applications

The main task of the designed product - SFIPF FW - is to separate the private network (Screened Subnet) from the public network (i.e. Internet).It was assumed that the protected network users are not the source of attacks. Screened Subnets architecture are four network segments designated to hosts only servers that need the access. Two routers (AR, IR) are the main part of the network architecture. Access router (AR) perform the functions: packet filtering with an analysis of the state of connections, control user access to the resources of the public network DMZ (Demilitarized Zone), defense internal network against attacks (DoS attacks, port scans), translation addresses and ports using N/PAT (Network / Port Address Translation) and acting as a gateway VPN.

Whereas interior router (IR) performs based packet filtering with the analysis of the state of connections, control user access to resources on the private network DMZ and the public network based on MAC addresses and has a function an intermediary proxy server for web services. There are many advantages of this concept, i.e.: create multiple subnets using a single switch, separation of terminals in the DMZ, access control and filtering traffic to the VLAN, because the traffic passes through the router, controlling traffic between VLANs and high scalability DMZ. Packet filtering uses the classic applications such as: Netfilter (IP Tables, Ipfwadm, Ipchains, OpenVPN and Proxy servers. These applications support VPN routing, Dynamic Host Configuration Protocol (DHCP).

AR router with SFIPF FW applications is connected to at least two networks, commonly LAN or WAN (Internet) or a LAN and PSTN's network. SFIPF FW applications control is based on a system of class Unix - Linux Kernel PLD (at least) 2.6. Kernel 2.6 with the security extension providing a low hardware requirement for computing power, many security module and software that has many features for supporting access control security policies. There have been developed filtering rules (script Bash_f1) using the Bash language for ARand IRrouters.Bash_f1 script contains a collection of files stored in the *etc/firewall* folder (Fig. 3.).

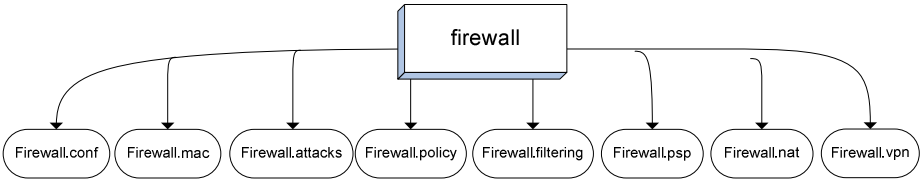


Fig. 3. SFIPF FW directory structure and important files paths explained

The rule in the bootstrap files can be modified by:

- protection against packet with a forged source address: */net/ipv4/conf/all/rp_filter*
- terminal defense against DoS attacks (SYN flood and Smurf): */net/ipv4/tcp_syncookies and icmp_echo_ignore_broadcasts*
- rejection of ICMP redirect and source route, which can alter the routing tables: */net/ipv4/conf/all/accept_redirects and accept_source_route*
- defense against bogus ICMP: */net/ipv4/icmp_ignore_bogus_error_responses*

The examples of IP packet filtering and NAT rules that they were implemented in AR Router:

```
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A INPUT -s 10.0.0.1 -d 10.0.0.1 -j ACCEPT
$IPTABLES -A INPUT -p udp -i $EXT_DEV -d $EXT_DEV_ADDR_1 –m multiport --dports 17003, 17004 -j LOG --log-prefix "VPN"
$IPTABLES -A INPUT -p udp -i $EXT_DEV -d $EXT_DEV_ADDR_1 –m multiport --dports 17003, 17004 -j ACCEPT
$IPTABLES -A INPUT -i tun0 -s 192.168.05.0/24 -p tcp --dport 22 –m state --state NEW -j ACCEPT
$IPTABLES -A INPUT -p icmp -j ACCEPT
```

More information about the Bash_f1 application is presented in Laskowski [4].

5 Secure Softphone Application

The second product, called SSPj, used to guarantee the confidentiality of telephone services (VoIP). The VoIP popularity stimulates creating new applications and evolution of existing developers' environments for the application of such type (user agents, servers, etc.). SSPj allows the effective implementation of technologies (speech and video codecs) and protocols:

- Session Initiation Protocol (SIP),
- Secure Real-time Transport Protocol (SRTP),
- Media Gateway Control Protocol (MGCP),
- Real Time Protocol (RTP).

Process of establishing connections and the SIP signalling is an important element based on the http protocol (HyperText Protocol Transfer) with four types of logical messages, where each has determined functions and communicating with remaining elements of the SIP system. Based on a detailed analysis an all libraries - PJSIP solution was chosen. PJSIP libraries are written entirely in ANSI C language and so language is also used at triggering the function, variables and structures of delivered data by PJSIP. The structural approach towards the paradigm of writing the software complicates the coding for the programmer, in the comparison from e.g. C++, especially at the cooperation with GUI object libraries. However, simultaneously it provides the high productivity for the written application and reduce RAM footprint than object libraries about similar functions. The expected properties of the VoIP service are proposed to be achieved by SSPj application and:

1) Network protocols: SIP, SRTP, MGCP and RTP protocols.
2) Advanced Encryption Standard AES CBC with 128/256 bit key.
3) Hash Message Authentication Code HMAC with iterative cryptographic hash functionSHA-2 in combination with a secret shared 256 or 512 bit key.
4) Public key infrastructure X.509 generated with OpenSSL.

Mechanisms of the security delivered by PJSIP are based on Secure RTP libraries libsrtp, of OpenSSL, and the function built in of the MD-5 abbreviation for basic giving authentication SIP news (Digest) (Fig. 4).

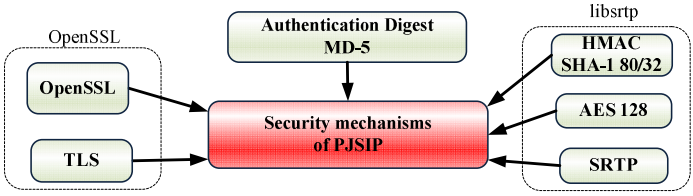


Fig. 4. Security delivered in the set of PJSIP

The interrelation of SSPj libraries being based on a layered and modular structure and the stream-oriented flow of data (Fig. 5.) let the PJSIP unlimited optimization.

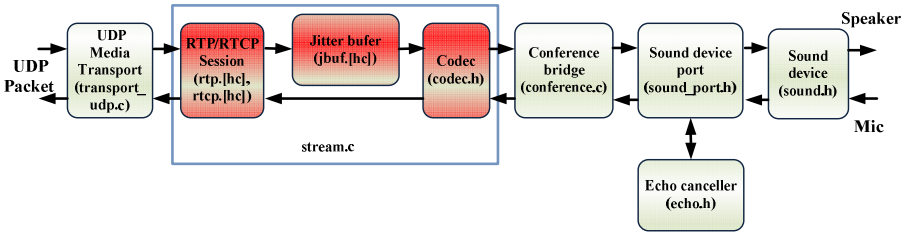


Fig. 5. Flow of voice data in PJSIP

The procedure of action of the protocol is shown in a figure (Fig. 6.). With main reason of using the algorithm AES-CTR there is a fact that the key can be generated before media data intended to encode is accessible. Thanks to that a needful time is reduced.

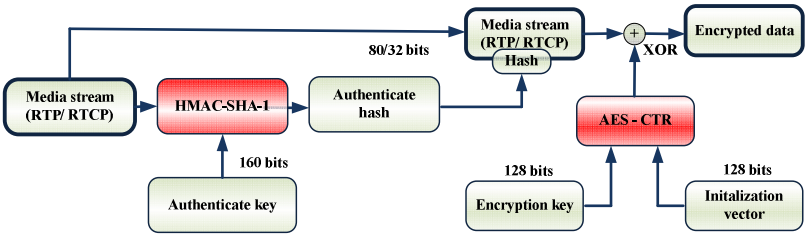


Fig. 6. Encoding and authenticating media data in the SRTP protocol and SRTCP

A constant delay of preparing packages is reduced. Another advantage of using the AES algorithm under the stream-oriented procedure, there is a lack of the need to level data to the full block what would increase of size of every package to a maximum for 15 bytes. The essential mechanism of SRTP giving authentication is based on an algorithm HMAC-SHA- 1. A content of authenticated fields of the package is subject to hash and 160 - bit secret code (auth-key). A 160-bit code which next is shortened is a result to 80 is optional, if reducing the size of the package is necessary, up to 32 bits. Thanks to applying above procedures it is ensured appropriate securities of the data transmission of the real time.

The most important element associated with the SSPj is the infrastructure of the public key which is designed by the OpenSSL package. With this platform of the security the certification centre (CA) will be built, the CA certificate will be formed with the Pearl script at applying guidelines of the standard of the X.509 certificate. The CA certificate can be signed by the commercial, public centre of the certification, however it requires expenses.More information about the Softphone application is presented in Jankowski [5].

6 Verifying the Confidentiality Requirements of the Test Bed

For the research aim the test bed and the real network was performed to enable a completion examine for couple of scenario:

1) Correctness: distress session data processing and functioning distinguish environments.
2) Compatibility with different solution for server platforms and terminals.
3) General understandable in talk with and without coding.
4) Resistance for selected attacks: signaling and media torrent tap, termination of session, impersonation a server, call hijacking, refuse a service D/RDoS (Distributed / Reflected Denial of Service).

The proposed of VPN should be constructed in a centralized topology based on Site-to-Site IPSec tunnels established between network departments using specifications given in publication

[6-7], reconfiguration and diagnosisnetwork based on the publication by [8-9] and RFC [10]. The table (Table 1) shows a comparison of the application during the selected attacks.

Table 1. Analysis of responses to selected attacks

Attack	Evaluation of security applications
Wiretapping signaling and media stream	– No possibility of network services such as data transport and VoIP. – Short-term increase in the load of network elements (AR, CA Server). – Reduction of throughput.
Cracking an IP session	– Transportation Data and VoIP are continued. – Bogus packets do not interrupt the application.
Server replacement	– Authorized user will not be able to log on to the server. – The error has been signaled in the terminal (PC, phone) and displaying the message: “Certificate invalidity or error”.
Callhijacking	– The attacker could not log in to the CA server, but entitled user - could. – Illegal attempt to registration has been signaled in the server and displaying the message: “User with unauthorized or invalid certificates”.
Distributed Denial of Service and other attack	– Correct operation of the application. – DoS, DDoS and DRDoS attacks been locked. – Incorrect incidents were recorded and displayed on the monitor server.
Efficiency IP filtering for attacks	– 99.9 percent attacks stopped. – Correct decoding for different lengths of the key and data.

The choice of this technology is justified by the need of building multiple connections through which will be providing an access to a variety of services, which in the case of the use of SSL would be very inefficient. The figure (Fig. 7.) shows the captured traffic to IPSec tunnel having misunderstood the content - the data is properly encrypted.

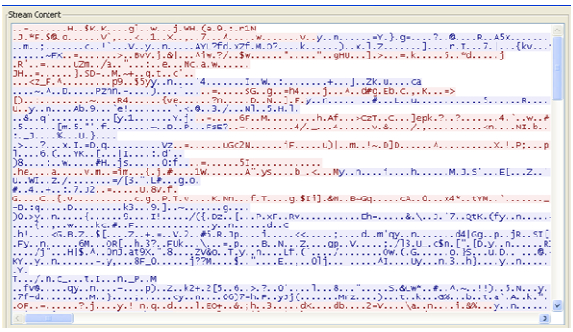


Fig. 7. Captured telnet communication VPN (IPSec tunnel)

Then, using IxChariot performance test was conducted IPSec tunnel. To this end, 10 times the size of the uploaded file 500 MB. This allows the number of times that the term high reliability of the test and averaging the result. In the first case sent

without an aligned movement of the tunnel. Average bit rate value obtained when estimating the test result was 92.38 Mbps Figure rate as a function of time assembled together without the tunnel is shown in the following figure (Fig. 8.).

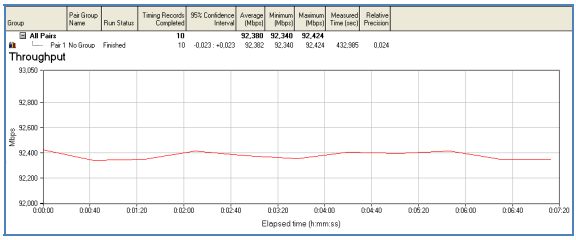


Fig. 8. Throughput without tunnel

However, after the statement of IPSec tunnel and carrying 10 times file transfer rate of the mean value was 76.13 Mbps The following figure shows the graph of the resulting average throughput as a function of time (Fig. 9.).After analyzing the charts (Fig. 8 and 9) it can be seen that the rate in the IPSec tunnel slept about 20%, which is the result of overhead, which introduce additional headers IPSec.

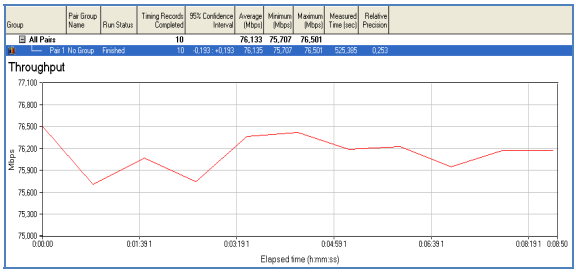


Fig. 9. Chart compiled rate of tunnel

The results of measurements present that SHA algorithms and AES (in the model of combining encrypted blocks) are characterized by a great efficiency of calculations (Fig. 10.). This productivity is completely enough for IP telephony applications, taking into consideration the fact that on the computer works other processes, seizing the memory and resources of the. As regard the flow of the stream in the real time (maximum to 64 kb/sfor the PCM codec) and of signalling, cryptographic algorithms will not increase the delay or jitter.

When length of the key or the hash is increasing the level of security is also increasing, but and the efficiency is reduced. The longer key causes that the algorithm is more robust to all sorts' attacks. Therefore it is legitimate so that in protocols of the protection algorithms are used with most possible longest key or the hash. In this case, for the protection of the telecommunications system of the strong AES 256 algorithms and SHA 256 or 512 is right test. It is established by security protocols before beginning the data transfer.

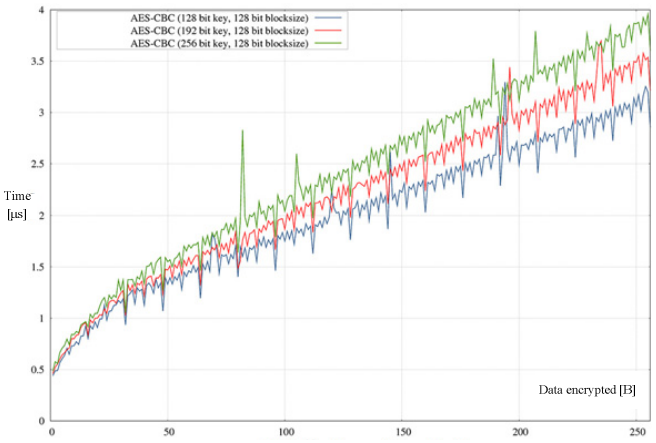


Fig. 10. The time's graph of encoding data for different lengths of the key and data

The figure (Fig. 11.) shows an example of changing the throughput of IP traffic at the attacks and enabled security mechanisms. The attack did not break the service (data, VoIP). Network throughput decreases by 10% for 1000 pps (packets per second) and 40% for 5000 pps under the high (80Mbps). The degradation throughput is at a satisfactory level. The degradation throughput is at a satisfactory level, as services are still performed. Service user has not felt the effects of the attack. The results of the other scenarios also demonstrated in the operation of the security mechanisms that are commercial integration and custom applications.

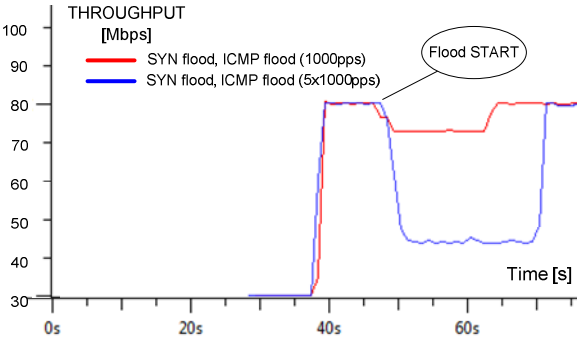


Fig. 11. The time's graph of throughput of IP traffic for SYN and ICMP flood with SFIPF FW and SSPj security mechanism

7 Conclusions

Network without the safeguards applied is very vulnerable. Attacks can be achieved by using open source tools (application or software). Easily you can also write your own programs that may pose a threat to network services. It is therefore necessary to

protect the data transport networks and other services. It is necessary to use security mechanisms to protect protocols and data exchanged between the network elements. There are many proven security technologies that can be applied. Among them were selected products with optimal properties, i.e. a secure protocol and virtual networks. IPSec and VPN enable integration with custom applications. These types of integration are desirable and will ensure the confidentiality of data in wide and local area networks.

Custom solutions are best fitted in the high security requirements. Security products do not require high computing power and advanced equipment and operating systems. Generally available routers, servers and libraries offer sufficient possibilities. The uniqueness of the commercially available products and knowledge programmer provides effective protection against attacks, as shown in the article. The essence of security is state full inspection and packet-filtering firewall simpleminded in router. Secure Softphone application using PJSIP library provides increased functionality.

SFIPF FW with Bash_f1script blocks most (99.9%) network connection attempts to decrease the chances of attacks (DoS and other attacks). A set of rules, known as a SFIPF policy, defines how the firewall switches outbound and inbound IP network traffic. Firewall is typically network devices with many different configuration settings but uses a dedicated rules and procedures. Maximizing state full inspection and packet-filtering firewalls efficiency is a balance between blocking and allowing application or programs access to the local and wide area network.

Secure Softphone application using PJSIP library was able to guarantee the security of VoIP services. PJSIP is an open source and popular multimedia communication library implementing standard protocols such as RTP, SIP. It combines signalling protocol with NAT traversal functionality into high level API and it supports data, audio and video. Security is based on Secure RTP libraries libsrtp, of OpenSSL, and the function built in of the MD-5 abbreviation for basic giving authentication SIP news.

Based on the results of experiments can be said that the firewall with VPN and softphone application function in real network (ie Internet) as required security. On the basis of tests it can be concluded that the application realize defined functionality. The application has been demonstrated resistance to attacks during ensuring confidentiality and effective distribution of certificates and asymmetric keys. The areas of softphone evolution are:

1) Adding support for protocol IPv6.
2) Development of application security in order to eliminate servers.
3) Implementation watermarking.

SSPj v. 2.0 is a "significant added value" potential area of adaptation application authentication correspondents on a "high" level - integration with the new multinational standard for secure voice and data communication SCIP (Secure Communications Interoperability Protocol), etc. The next stages of experimental research will concern new software and hardware modules e.g. new embedding and extracting methods based on multimedia content watermarking [11] as well as subscriber authentication procedures based on Gaussian Mixtures Models [12].The new software SSPj v.2.0 with new built-in modules will guarantee advanced level of confidential and integrity data.

Presented structure of the firewall and softphone applications should offer increasing level of the safety in heterogeneous environment. Used libraries create potential capabilities of the optimization of chosen attributes safeties for applications in universally used mobile terminals of the cellular GSM/UMTS network operators. It gains the special significance for future solutions of the network of a NGN next generation (Next Generation Network) multiplexing of solving optical DWDM technologies (Dense Wavelength Division Multiplexing) and radio communication of the third generation - of HSDPA technologies offering the extended pallet of multimedia services to UMTS through the implementation LTE (Long Term Evolution).

References

1. Siergiejczyk, M., Krzykowska, K.: The analysis of implementation needs for automatic dependent surveillance in air traffic in Poland. In: Weintrit, A. (ed.) TransNav - The International Journal on Marine Navigation and Safety of Sea Transportation, pp. 241–245. CRCPress / Balkema Taylor & Francis Group, London (2013)
2. Piotrowski, Z., Nowosielski, L., Zagozdzinski, L., Gajewski, P.: Electromagnetic compatibility of the military handset with hidden authorization function based on MIL-STD-461D results. In: Proceedings of the Piers 2008 Cambridge. Book Series: Progress in Electromagnetics Research Symposium, pp. 116–120 (2008)
3. Nowosielski, L., Wnuk, M., Ziolkowski, C.: Interlaboratory Tests in Scope of Measurement of Radio Disturbance. In: European Microwave Conference, Rome, Italy, vol. 1-3, pp. 288–291 (2009)
4. Laskowski, D., et al.: Filtration efficiency of traffic IP, ZNS'14, Poland (2013)
5. Jankowski, D., Laskowski, D.: Increasing the robustness of the telephone service of the heterogeneous environment with using open source solutions. In: Dependability and Computer Systems - DepCoS-RELCOMEX 2012, pp. 37–52. WUT, Wroclaw (2012)
6. Bajda, A., Wrażeń, M., Laskowski, D.: Diagnostics the quality of data transfer in the management of crisis situation. Electrical Review 87(9A), 72–78 (2011)
7. Lubkowski, P., Laskowski, D.: The end-to-end rate adaptation application for real-time video monitoring. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 295–305. Springer, Heidelberg (2013)
8. Chudzikiewicz, J., Zieliński, Z.: Reconfiguration of a processor cube-type network. Electrical Review (9), 139–145 (2010)
9. Kulesza, R., Zieliński, Z.: Diagnosis resolution of processors' network using the comparison method. Electrical Review (9), 157–162 (2010)
10. RFC 2401 Security Architecture for the Internet Protocol, Obsolete by RFC 4301, RFC 2411 IP Security Document Roadmap, RFC 4301 Security Architecture for the Internet Protocol, IETF Working Group (2014), `https://www.ietf.org/rfc.html`
11. Lenarczyk, P., Piotrowski, Z.: Parallel blind digital image watermarking in spatial and frequency domains. Telecommunication Systems 54, 287–303 (2013)
12. Piotrowski, Z., Wojtuń, J., Kamiński, K.: Subscriber authentication using GMM and TMS320C6713DSP. Electrical Review 88, 127–130 (2012)

Test of the Multimedia Services Implementation in Information and Communication Networks*

Piotr Łubkowski and Dariusz Laskowski

Institute of Telecommunications, Faculty of Electronics,
Military University of Technology Gen. S. Kaliskiego 2 Street, 00-908 Warsaw
{plubkowski,dlaskowski}@wat.edu.pl

Abstract. The paper presents results of the analysis of impact of numerous network properties on the functionality of the environment that provides multimedia services. Due to the multiplicity of characteristics and parameters we propose to restrict the set of determinants to the packet end-to-end delay, throughput, jitter, packet losses, etc. On the basis of the literature analysis and our own experience the mathematical relation was derived. It allows estimating the functionality of network services for TCP / IP stack. Performance tests were carried out for the most popular applications in the environment that reflects the realistic conditions of the Wide Area Network. The results obtained made it possible to determine the values of the indicators for the proposed equation.

Keywords: multimedia services, IP network, performance tests.

1 Introduction

A network environment includes commonly operated information and communication networks which are part of a telecommunication network. It constitutes a framework for a platform generally used for provision of various multimedia services. Increase in computing power of processors implemented in network stations stimulates the increase in a set of different points of access to generally accessible wide area network resources. At the same time, the network environment opens for evolutionary changes in the process of telecommunication services performance [1].

Today, the representative of the modern society uses in its everyday life more and more technical "innovations" such as technology advanced phones, pockets, smart phones and tablets. As part of the network operator service package, they offer numerous services. Except for the possibility of access to web sites, e-mail and transfer of files, the telephone services covering image and sound transfer gain increasingly more interest. Thus, there is an economically justified need to evaluate the functionality of the environment that provides media services that are the result of numerous network properties.

* The work has been supported by the European Regional Development Fund within INSIGMA project no. POIG.01.01.02,00,062/09.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_31,

2 Functionality

The functionality of services may be presented from different perspectives. One of them is focused on practical use and depends on such components as availability, accessibility and continuity. Additionally, the functionality of services may be affected by a number of other properties such as intuitive use, quick operation etc. [2].

Considering the conclusions drawn from both the experience related to operation of real network environments and the analysis of published scientific works as well as standardization documents, for the purposes of functionality assessment, it is necessary to limit the set of factors considered to only the most important, i.e. to delays in delivering packets, data flow capacity, delays fluctuations, packet loss, packet duplication, change of delivered packets sequence, error rate. Furthermore, elements that the user has direct contact with are also considered, such as the number of essential service applications and the number of network access points.

The condition for achieving and ensuring functionality of services is the stability and correctness of the hardware and software platform containing network components of data transfer (i.e. telecommunications switches, backbone routers, wire and wireless links etc.) and providing services (i.e. servers). An important determinant is the proper configuration of devices as well as system and application software. Various applications have different network requirements, e.g.:

1. Audio and video transmission requires small delays but tolerates partial packet loss.
2. Transfer of data does not require continuity of transmission, however all packets must be delivered using a large bandwidth.
3. In the case of instant messengers, a transmission without delays must be ensured and large bandwidth is not required.

It can be noticed that the diversity of applications and devices used by network users forces the service providers to ensure time-variable network availability. The last element that may have an impact on services functionality is the technical aspect, i.e. the place and method of connection with network and network equipment (and its redundancy) used by the operator. Wire connection ensures sufficient bandwidth and stability, whereas wireless connections provide mobility. Considering the above it may be concluded that the random and deterministic events having an impact on individual parameters determining the resultant functionality of services that may be classified based on the general mathematical dependency:

$$F(t) = (P1 \cdot p) + (P2 \cdot o) + (P3 \cdot j) + (P4 \cdot u) + (P5 \cdot d) + (P6 \cdot k) \quad (1)$$

where:

- F: functionality,
- p: bandwidth (data channel),
- o: the inverse of packet delivery delay (round trip time),
- j: the inverse of jitter,
- u: the inverse of percentage packet loss,
- d: the inverse of packets duplication,
- k: the inverse of change of packets sequence,
- P1,2,3,4,5,6: weight variables specifying the influence of a given parameter.

Due to the fact that the functionality is not expressed in units of measure, therefore the parameters used are represented by their absolute values. The above formula was created as the outcome of our experience gained while the multi-aspect analysis of network events occurring in real and laboratory network environments developed as part of numerous research and implementation projects [6,7]. Furthermore, the conclusions drawn from the analyzed literature were also included [8,9,10]. Due to the simplification, at this research stage, it does not take into account other components such as the impact of external conditions, i.e. the type of application used etc. The above mathematic dependency (1) will be specified by assigning concrete values to the weights Pn ($n=1…6$) based on the conducted tests.

3 Research Environment

The research environment will be developed in accordance with the postulated properties in order to reflect the essential real characteristics of telecommunications networks. Input data and criteria are based on delay values and packet loss for telecommunications services referred to in International Telecommunication Union-Telecommunication Standardization Sector [3÷5]. The more important parameters are presented in the below table (Table 1).

Table 1. Acceptable QoS values for services related to data transfer [3]

Medium	Application	Degree of symmetry	Typical amount of data	Key performance parameters and target values	
				One-way delay/page	Delay variation / Information loss
Data	Web-browsing HTML	Primarily one-way	~10 KB	Preferred <2s Acceptable <4s	
Data	Bulk data transfer / retrieval	Primarily one-way	(0,01÷10) MB	Preferred <15s Acceptable<60s	N.A. / Zero
Data	Usenet	Primarily one-way	> 1MB	Can be several minutes	

If the above requirements are met, the user thinks that the network fulfills expectations in terms of objectivity. Additionally, the figure (Fig. 1) shows the assumed acceptable values of packet delay and loss presented in recommendation G.1010 [3].

A various and diversified research methodologies are possible, which are used to determine the functionality of this sort of complex technical objects. The plurality of solutions depends on the knowledge of people conducting research as well as hardware and software possibilities. Thus, the telecommunications network model reflects the most important elements of a real network, in view of the implementation ability and functionality of network services. The developed analysis concept takes into account numerous components determining proper implementation of network services so as to enable performance of complex scenarios and ensure reliability of output data.

Considering the above generalized specificity it is proposed to assume the following requirements as regards the research environment (ground/test environment) comprising projects that are requested and necessary to ensure reliability. They are referred to as:

1. Ensuring cooperation of operation, commercial and open source systems [7].
2. Ensuring diversity of applications and services used [8,9].
3. Ensuring reliability of the developed model (input data) [10].
4. Ensuring exchange of:
   - hardware and software equipment:
     - program: PC computer (3 Fast Ethernet cards),
     - hardware of a renowned company: router including IP Services software, access switch of II layer.
5. Ensuring correctness of applications for generation and analysis of data:
   - emulator of Candela Technologies LANforge network,
   - device for testing applications in real conditions IxChariot,
   - advanced analyzer of Wireshark packets.
6. Ensuring connection of the working station to the created network: UTP twisted pair cable, 5e category and 802.11n standard.

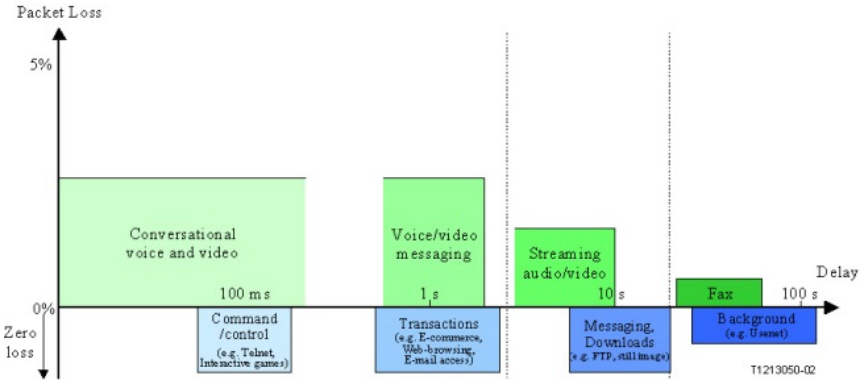


Fig. 1. Acceptable values of packet delay and loss for G.1010 standard [3]

An important element for testing the functionality of services is the selection of network environment and adequate set of test software. Effective applications LANforge GUI, Wireshark, IxChariot enable obtaining a sufficient set of resulting data for reliable presentation of essential aspects of network environment.

Due to the research diversity and problems with the identification of changes of network parameters during services performance and their monitoring in real time regimes, it was decided to limit the research to verification of packet delay effects and its fluctuations to packet transport. This testing method is possible using LANforge, IxChariot (Fig. 2). Additionally, LANforge GUI enables an overview of current instantaneous parameters values such as bandwidth use and packet loss, as well as delays and other.

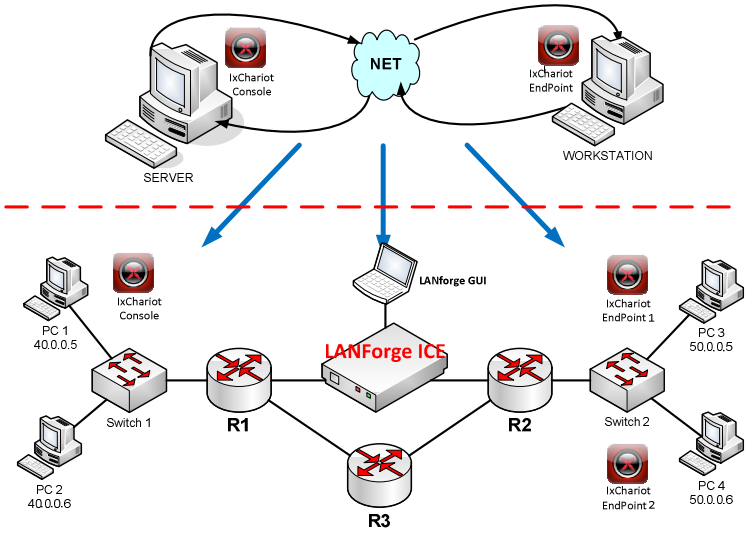


Fig. 2. Generalized architecture of research environment

4 Research Results

The research was performed in accordance with the adopted test scenarios, all of which will be presented in the article. They will include the influence of delay variations on services which are popular and commonly used for commercial and non-commercial purposes, such as:

1. File Transfer Protocol.
2. World Wide Web browsing using http protocol.

The simplified testbed architecture is shown on Fig. 3. When sending and receiving packets, the measurements were performed at the maximum available bandwidth of 10 Mb/s. Diagrams created as a result of the tests using the IxChariot application present the maximum amount of sent data in Mb/s for services with given parameters such as the size of the file (Fig. 4).

Fig. 3. The simplified testbed architecture

The ftp file transfer service (Fig. 5) is presented below. At the same time, file "packages" with a size of 1 MB (green color in the diagrams) and 100kB (red color) were sent.

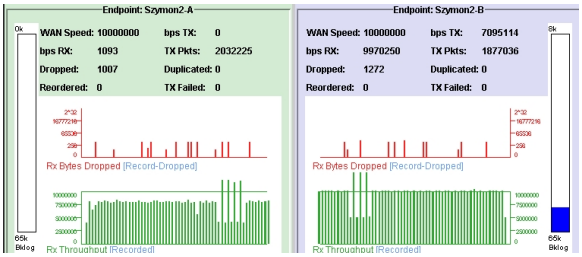


Fig. 4. Preview of current ftp services properties using the LANforge GUI application

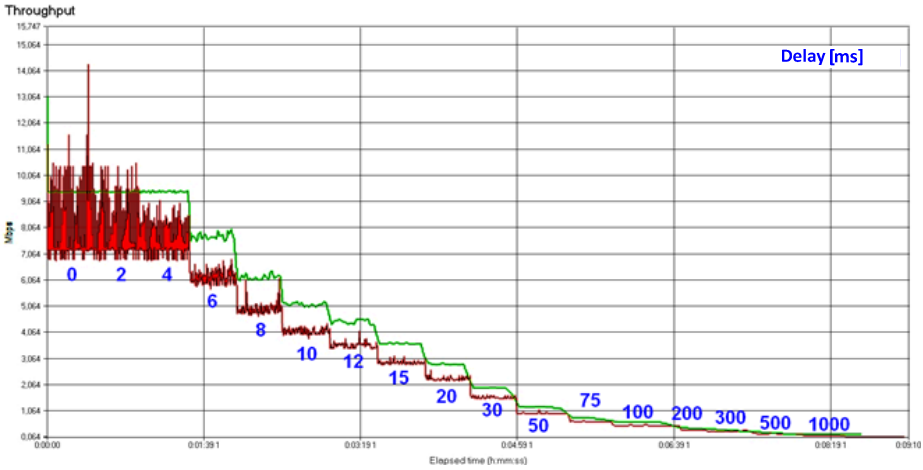


Fig. 5. Impact of delay on the file transfer service

Based on the data from the graphs it can be noticed that (Fig. 4, Fig. 5):

1. Transfer of a smaller amount of larger packets ensures steady use of the entire available bandwidth, whereas transfer of a great amount of small packets results in unsteady use of the bandwidth. In conclusion it should be noted that at the time of sending the large file size of 400MB (400x1MB) in fact uploaded are small files of 300MB (3000x100kB).
2. Ftp service is sensitive to loss of sent packets. Even the losses below 0.1% of packets cause problems with the service implementation, because lost packets must be retransmitted, which results in additional transmission delays. The transferred file must not be delivered in parts, otherwise, in many situations it will be impossible to use it.

Another parameter that affects the functionality of file transfer is the jitter. This parameter was tested in four time intervals (1%, 10%, 50%, 100%) of delays with a maximum value of 10ms, 25ms, 75ms, 150ms and 500ms. The results are presented below (Fig. 6).

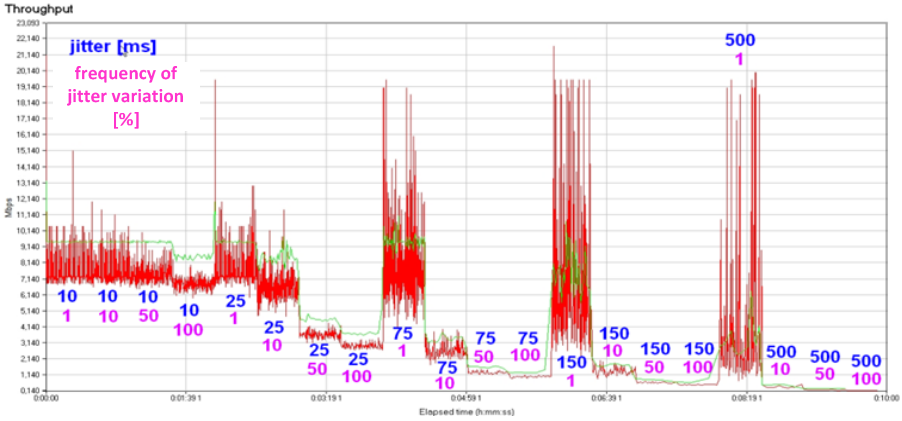


Fig. 6. Impact of delay variation on the file transfer

For instance, 10%-occurrence of a delay with a maximum value of 50ms means that one out of the ten transmitted packets will be delivered with a delay ranging from 0 to 50ms. Thus, it can be seen that jitter affects the service functionality similarly as the delay parameter, but with a less significant effect. In a real wide area network, jitter occurs almost at all times, because it involves variations that take place in the network on an ongoing basis and depend upon them.

Based on the presented diagram, it can be stated that the frequency of few percent delay variations is acceptable and does not significantly affect the service performance. In the case of several dozen percent delay variation of several hundred ms, the impact of the service is negative, similarly as the delay. Jitter has a very adverse impact on the performance of real time services. To prevent it, buffering of transferred data is applied, which, in turn, involves additional delays. The results obtained were the resultant of repetitive output data acquired on the basis of research carried out in a numerous scenarios. Representative data were presented in basic diagrams (Fig. 4, Fig. 5, Fig. 6) and the key findings were specified in conclusions, and used to determine numerical weight in formula (1) concerning the tested functionality:

$$F(t) = (0{,}4 \cdot p) + (0{,}1 \cdot o) + (0{,}05 \cdot j) + (0{,}3 \cdot u) + (0{,}01 \cdot d) + (0{,}14 \cdot k) \quad (2)$$

Although the proposed factors were obtained as a result of a conducted scenarios, however, at the present stage of the research presented formula cannot be treated as the final result. In order to achieve full credibility of the proposed equation further research and estimations has to be undertaken.

5 Conclusions

In the era of development of techniques and technologies applied in wide area networks (i.e. Internet), the requirements for both the proper functioning and effective use of the possessed operational potential by the service provider (operator) increase as well.

The more important determinant for both the user and service provider is the bandwidth, as it directly influences the "comfort" of operation, and thus the functionality of multimedia services used. Unfortunately, the network problems, i.e. delays and jitter often have essential and degrading impact on data transfer and target service performance. In further tests, it is planned to analyze the impact of packet loss, duplication and change of packets sequence. Due to the time-varying demand for services it seems to be necessary to have knowledge on forecasting, with an adequate advance, of the need for access and transport resources enabling delivery of the required amount of data to the user (subscriber).

That can be achieved by time-continuous identification of network environment functionality. Therefore, using the available knowledge and programming tools, one of the numerous, possible to be specified, mathematical dependencies characterized by transparency was presented. It is obvious that its components can be specified for other topologies taking into account the adjustment of weights for the next network architecture tested.

Acknowledgement. The work has been supported by the European Regional Development Fund within INSIGMA project no. POIG.01.01.02,00,062/09.

References

1. Lubkowski, P., Laskowski, D.: The end-to-end rate adaptation application for real-time video monitoring. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 295–305. Springer, Heidelberg (2013)
2. Coverdale, P.: ITU-T Study Group 12: Multimedia QoS requirements from a user perspective (2010)
3. ITU-T G.1010, End-user multimedia QoS categories
4. ITU-T Y.1241, ITU-T Recommendation Y.1241 - Support of IP based Services Using IP Transfer Capabilities
5. ITU-T G.1541, A Basis for IP Network QoS Control and Traffic Management
6. Laskowski, D., Lubkowski, P., Kwasniewski, M.: Identification of suitability services for wireless networks. Electrical Review 89, 128–132 (2013)
7. Laskowski, D., Bylak, M.: Efficient diagnostics encoding mechanism for wireless networks. Electrical Review 89, 133–138 (2013)
8. Nowosielski, L., Wnuk, M., Ziolkowski, C.: Interlaboratory Tests in Scope of Measurement of Radio Disturbance. In: 2009 European Microwave Conference, Rome, Italy, vol. 1-3, pp. 288–291 (2009)
9. Bajda, A., Wrażen, M., Laskowski, D.: Diagnostics the quality of data transfer in the management of crisis situation. Electrical Review 87(9A), 72–78 (2011)
10. Siergiejczyk, M., Rosiński, A., Krzykowska, K.: Reliability assessment of supporting satellite system EGNOS. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 353–363. Springer, Heidelberg (2013)

Unified Approach to Network Systems Multicriteral Analysis

Jacek Mazurkiewicz

Institute of Computer Engineering, Control and Robotics,
Wroclaw University of Technology ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
Jacek.Mazurkiewicz@pwr.wroc.pl

Abstract. The paper is focused on the methods of network services exploitation. The approach is based on two streams of data: dependability factors and the features defined by the type of business service realized. The dependability means the combination of the reliability and functional parameters of the network. The proposed method is based on modeling and simulating of the system behavior. To simplify integration of the model author propose an automatic solution that is integrated with the tools chosen for system analysis. Analysis is done with a usage of open-source simulation environment that can be easily modified and extended for further work. Based on the simulation results, some alternatives can be chosen in case of system or service failure. This way it is possible to operate with large and complex networks described by various - not only classic – distributions and set of parameters. The results are converted to the unified system description and generic model. The model can be used as a source to create different measures – also for the economic quality of the network systems. The presented problem is practically essential for organization of network systems.

Keywords: network systems, reliability, dependability modeling.

1 Introduction

The contemporary network systems are created as very sophisticated products of human idea characterized by the complex structure. On the other hand the systems combine two types of resources: technical (engineering stuff) and information (algorithms, processes and management procedures). The systems are human-controlled and computer-aided devices. The reliability parameters of the system resources are at very high level – so the exploitation analysis of contemporary systems needs adequate models and calculation methods [24, 29]. In the era of e-shopping, e-banking, e-learning and e-services, Internet accessibility cause growing numbers of users and their needs. Trends of service personalization increase these requirements. To satisfy these needs, various concepts are proposed. One of them is a Service Oriented Architecture (SOA) concept, but the complexity of these systems, as much as their requires of dependability and management issues still need an improvement. This paper focuses on the complex network systems, where business service aspects are crucial for

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_32,

their owner (service provider). Unavailability of these systems causes large financial consequences and loss (not only in a sense of economic, but also as loss of a good name of the brand). In area of monitoring and still specified and in most cases commercialized. In fact these solutions (tools, models) are so specified for the implementation, that they could not be used easily for any other design or even slightly more complex one.

Moreover in the area of description and measurement dependability and functionality aspects are treated as separate categories but not as a hybrid method. For example, in case of analysis, metrics are used only for a business level of abstraction or system infrastructure level. Hybrid metrics (for both levels) are still under research [8, 25].

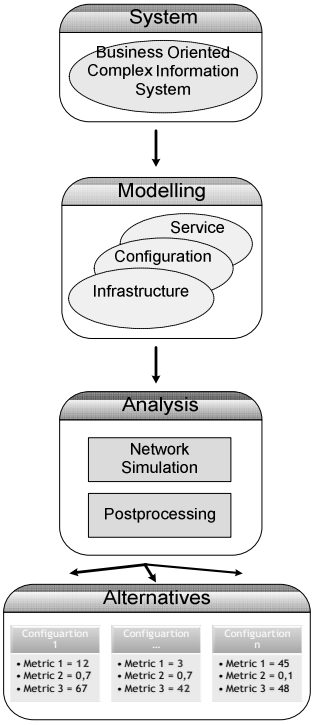


Fig. 1. Research concept - overview

In this paper we based on functional and dependability approach related with tree main parts: modeling, analysis and synthesis (Fig. 1). During more than 60 years the reliability theory was altered from the reliability of single and separated objects (elements) considered only two states ("efficient work", failure) to the contemporary dependability of systems or even the dependability of service nets. The indicated development of the reliability theory is the consequence of expanding the event sets taken into consideration for the reliability models. The present system dependability theory considers not only classical reliable events (failures or repairs) but tries to combine all types of the faults generated by the system resources (hardware,

algorithms, human-factor) and the environmental features which may disturb the operable state (attacks – for example). The main goal of the system exploitation analysis is to convert the discussion focused on the reliability function of elements (or structures created by the element sets) into the task performance or efficiency estimation. The tasks are realized according to the system services [20, 21]. The classical models used for reliability analysis are mainly based on Markov or Semi-Markov processes [3] which are idealized, it is hard to reconcile them with practice and is insufficient in general. We suggest the Monte Carlo simulation [17] for proper reliability and functional parameters calculation. No restriction on the system structure and on a kind of distribution is the main advantage of the method [18].

We call the approach as the functional-reliability models of network system exploitation. The computer systems analysis is the root for our elaboration but we believe it is useful for modeling of the wider spectrum of systems which realize tasks based on fully or partially available resources. We think about a discrete transport system or power management systems for example.

The computer and software equipment allows making the exploitation analysis more sophisticated. The simulation technique is the real chance to operate with large systems – where the number of elements is significant. The elements can be described by different sets of features. We can observe – in parallel – large number of events in quite long time-periods. This way we can collect data sets to very detailed presentation of the system life. Based on the data we are able to elaborate the formal theoretical approach to the network system exploitation [21, 22, 24]. Of course it is necessary to eliminate all these features which are very system-depend and not enough generic.

2 Network System Overview

Computer Information System (CIS) is described [19] as a 4-tuple:

$$CIS =< Z,HS,M,K > \tag{1}$$

where:
Z – tasks, HS – technical infrastructure (hardware, software, links), M – clients, K – chronicle of the system (understood as time functions of the system).

Since we propose to analyze the Information System from a business service perspective called *Business Service Information System (CISB)*, we have extended (1) and define *Business Service* (*BS*) as a set of business logic, that can be loaded and repeatedly used for concrete business handling process (ticketing service, banking).

$$CISB =< Z, M,BS,HS,K > \tag{2}$$

Business Service (BS) can be seen as a set of service components and tasks that are used to provide service in accordance with business logic for this process. Business service components which consist of a set of activities that are the lowest observable entities level (requests and responses).

$$BS_i = \bigcup_j SC_j ; j \in N_{BS_i} \tag{3}$$

Service Component (SC_i) is a service located on defined host (server) that determined service behavior, possibilities and requirements (i.e. authentication, data base service, web service, etc.). One host can have more than one service component.

$$SC_i = \bigcup_j T_j ; j \in N_{sc_i} \tag{4}$$

Technical Infrastructure (HS) is considered as a set of hosts and computer network and is assumed to have the aspects of TCP/IP traffic are negligible. Each host is described by server name (unique ID), host performance parameter and a set of technical services (i.e. apache web server, MySQL database). *Chronicle of the system* (*K)* are the time functions on each level of abstraction. *Clients* (*M*) consists of a set of users where each user is defined by its allocation (host), number of concurrently ruing users of given type, set of activities (a sequence of task calls - name of task and a name of service component) and inter-activity delay time (modeled by a Gaussian distribution). *Tasks* (*Z*) are the input data specified by the clients in case of business service usage (i.e. selection of a service imply its components with a specified choreography).

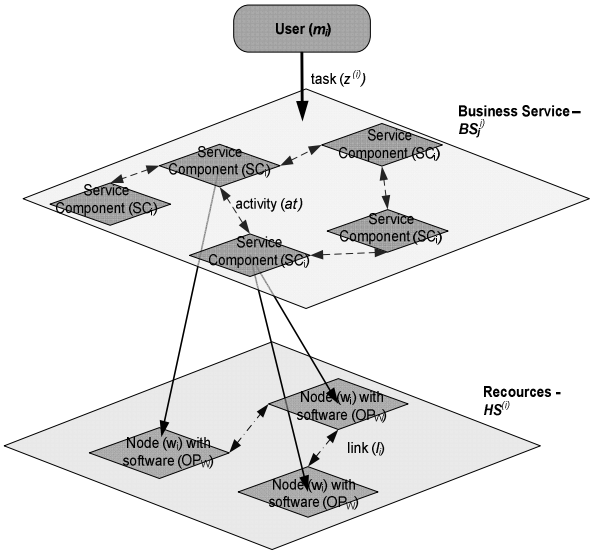


Fig. 2. Business service oriented information system – levels of abstraction

3 Approach to Analysis

There are various methods of system analysis. Some researches try to do it using graphs [9], others choose some simulation techniques. In this paper we consider system behavior that will be as close to reality as is can be. Usage of simulator allow us to mimic the behavior of a system. In literature, two main types of simulators can be found: a continuous time and discrete event based simulation [12, 22].

Continuous simulation requires a representation of the system using differential equations [9]. This type of simulator is predominately related with electric power studies. For this reason it will be excluded from further research. The other group of simulators are discrete events that describe the system behavior as a series of events. Classically discrete event simulators are basis for telecommunication and IT analysis tools. The set of the most popular simulators of this kind is as follows: *OPNET* [5,18] and *NS-2* [14], both well known by stakeholders, as well as *QualNet* [1], *OMNeT++* [7], *SSFNet/PRIME SSF* [23], and *SGOOSE* [9].

Experiments reported in this paper were performed using the *SSFNet* simulation environment developed by the Renesys Corporation with support from *DARPA*. *SSFNet* has large number of protocols models and network elements; moreover open-source code allows modification. In this paper Java based version of *SSFNet* was used since Java language allowed much faster development then a usage of C++.

SSFNet simulator consists of three major parts: *SSF* engine, *DML* language [8] and *SSFNet* models. The *SSF* (*Scalable Simulation Framework*) is public-domain standard for discrete-event simulation implemented in C++ with Java and C++ interface. Scalable Simulation Framework is a base for higher level - the *SSFNet*. *SSFNet* module is a collection of Java packages for modeling and simulation of networks and Internet protocols. Moreover *SSFNet* uses public-domain standard called *DML* (Domain Modelling Language) to configure simulation scenarios.

For the purpose of this work some extensions were developed, mainly connected with support for traffic generation (models of user behavior), simulation of business level services, implementation of resource consumption for requests processing. Since fault and failures models are integral part of dependability analysis the *SSFNet* was extended to in-corporate errors. Errors were introduced in different levels beginning from link failures, network adapter failures to software component failures [23]. Additional modules of the tool required the extension of its input language (*DML*) used in standard *SSFNet* version, but the most important extension was implementing Monte-Carlo approach [11] based on running simulation several times and calculating results based on averages values. In this way - during each simulation - the parameters described in by stochastic processes - were the traffic generation which modeled user behavior in a random way. They have different values (according to set-up distributions) including an influence on the system behavior. The capability of multiple runs of simulation was added to standard *SSFNet* package by changes in several *SSFNet* classes (setting up random seed and clearing all static collections). Results of simulation are recorded in specified output file that allows further post- processing in case of dependability metrics.

For the needs of this research, we provided two metrics of information system dependability - availability and response time. Due to a randomness of a user behavior the calculation of these metrics was done based on Monte-Carlo approach by repeating simulation of the same system N times over analyzed period T. Therefore, all defined below metrics are calculated as an average over all batches of simulation.

The availability function [3] $A(t)$ for the system is a probability that system is working properly in time t:

$$A(t) = P\{system\ is\ working\ in\ time\ t\} \tag{5}$$

In Business Service Oriented Complex Information Systems with more than one level of abstraction, we can suppose, that the system is available as a probability that in time t all requests come from users to the system and services are supported correctly. On this basis we can estimate that, the *business service availability* (*BSA*) can be computed on the basis of observed system uptime in the analyzed period T over N simulation as:

$$\boldsymbol{BSA} = \frac{1}{\boldsymbol{NT}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{t}_{\boldsymbol{up}}^{\boldsymbol{i}} \tag{6}$$

whereas t^i_{up} is a time of service being working in i-th simulation.
Above formula (6) requires defining what does it mean that service is working. Since we are looking on the system from the client perspective, we assume that service is working if and only if it responds to the client with a proper response. The downtime starts when for some request there is no proper response (the time of starting of response is used). It finished when for any request there is a proper answer (also a request send time is used in this case) [25]. In a very similar way we can calculate availability of the server (*SA*) as:

$$\boldsymbol{SA} = \frac{1}{\boldsymbol{NT}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{t}_{\boldsymbol{up}}^{\boldsymbol{i}} \tag{7}$$

In this case we can calculate another level of abstraction in *CISB*, that is hardware one. In this case, t^i_{up} is a time of server being working in i-th simulation.

Response time in a *CISB* is a time to pass between request to the system and correct response. In case of *CISB* it depends from network recourses, infrastructure capabilities and system overload. Proposed metric analysis the *business service response time (BSRT)* and is intended to be a numerical representation of client's perception of particular service components quality. It is calculated for each tasks separately as an average delay between the starting time of user response ($t_{i\_request}$) and getting answer ($t_{i\_response}$) from the service (i.e. only requests that were properly answered are taken into account).

$$BSRT = \frac{1}{N_{request}} \sum_{i=1}^{N_{request}} (t_{i\_request} - t_{i\_response}) \tag{8}$$

BSRT is considered to be very useful for ranking system configurations.

4 Environment Integration

Usage of *Domain Modeling Language* in *SSFNet* simulator causes a need of integration. In spite of the fact, that *DML* is simple text format it is difficult to read and write larger number of network nodes for simulation. Moreover extensions done in *SSFNet* entailed *DML* modification and *SSFNet* output format. Since *DML* text format is easy to transform to *XML* format, we propose an extended simulation output – called *XDML*. Creation of *XDML* languages give as many processing possibilities. First of all we can translate any language to an input format of this analysis module (Fig. 3). Secondly an *XML* format is easily processed by Java and *XML* techniques (i.e. *XSLT*, *DOM*, *SAX*, *JAXB* technique) which is helpful in creating model (in our case Information System) visualization: showing the structure of the network and it's element. Each network element has several functional parameters and user can graphically edit this information. It is worth to mention that every *SDL* device type is translated into suitable *XDML* one and every service description is interpreted and translated into service components and tasks. Moreover in proposed framework user is able to put its own variables and attributes based on *XDML* specification or use extend models (i.e. consumption model, operational configuration model) to simplify its work.

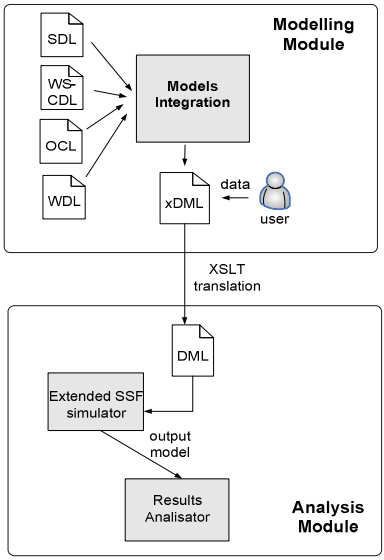


Fig. 3. Integrated Environment

Since analysis model is mainly based on Java interface of *SSFNet*, it helps to integrate *SSFNet* environment with created postprocessing module. It allows adding some additional features concerning Monte-Carlo simulation (i.e. progress bar) and a module responsible for plotting calculated metrics.

5 Case Study

For the case study analysis we propose an exemplar service system illustrated in Fig.4.

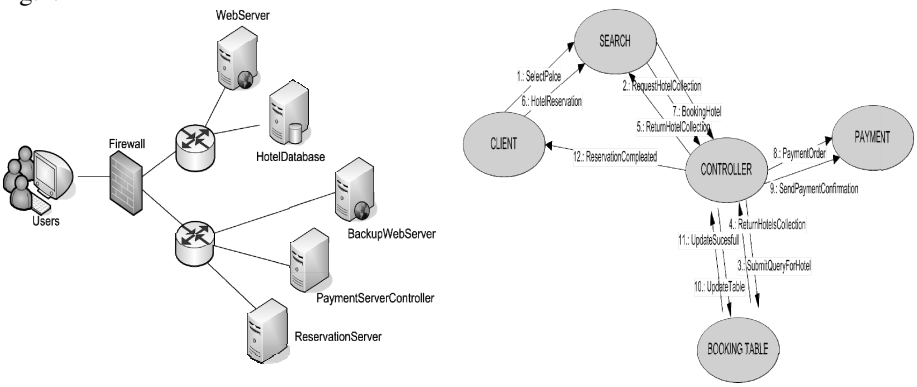


Fig. 4. Test system

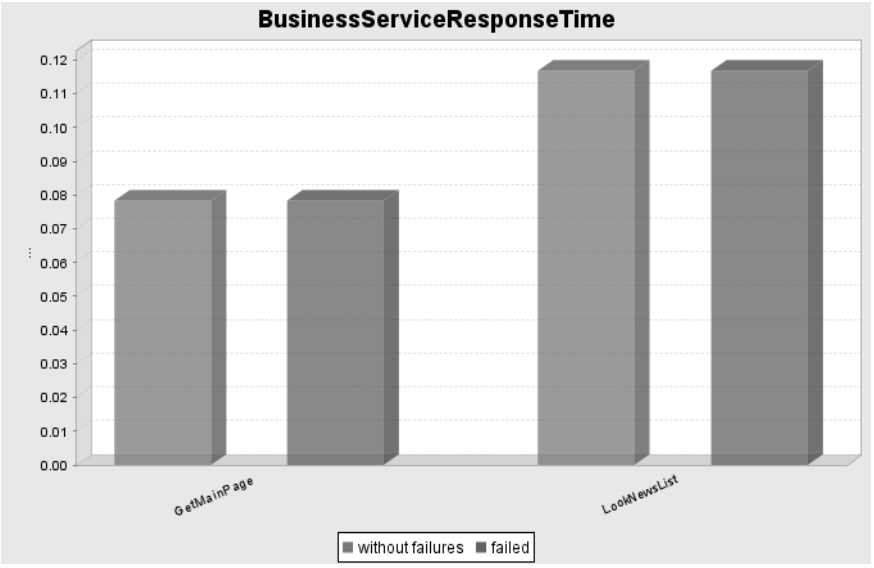


Fig. 5. Business Service Response Time - metric results

We have analyzed a system which consists of three networks: one is a client network, other service provider networks (secured by a Firewall). System is realizing simplified hotel booking system that allows booking an available apartment in hotel. Few servers are used for a proper service realization: *WebServer*, *HotelDatabase*, *ReservationServer*, *PaymentServer*, *Backup*. The service and its choreography is

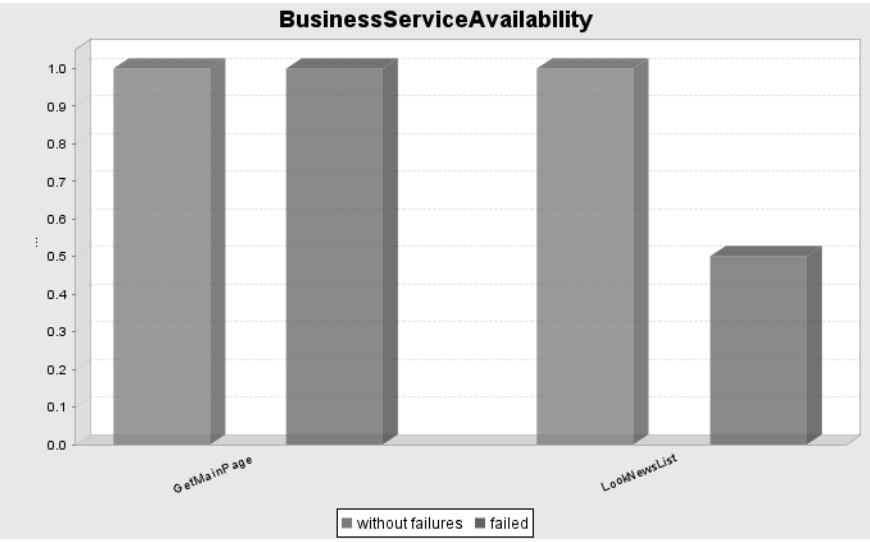


Fig. 6. Business Service Availability - metric results

described in Fig. 4. First of all, place of the hotel is being searched, than reservation is being made. At the end of this scenario payment is done with an interaction with given payment system [18]. Essentially the testbed system implements two main service scenarios: "*GetMainPage*" – of hotel reservation system and "*LookNewsList*" – as a final proof of payment. Each scenario is described using specified set of service component and interaction between them. Since simulation allows to observe different parameters of observed model we focused mainly on a metrics proposed in the previous section. In order to perform some interesting experiments we consider two hypothetic situations that may occur in our testbed system: first, that all elements are working properly, second some failures were introduced (the failure of *HotelDatabase* staring at 1000 s and finished at 5000 s.). The results of a comparison of the two configurations (without failures and failed) are given in Fig. 5. and Fig 6. The simulations was performed for simulation time of 10000 seconds and repeated 100 times. It shows that in case of failures performance and availability drops down for every level of abstraction.

6 Conclusions

We have presented a unified, abstract, formal model for modelling of network systems exploitation problems. Based on the results it is possible to create different metrics to analyse the system in case of reliability, functional and economic case. The metric could be analysed as a function of different essential functional and reliability parameters of network services system. The presented approach - based on two streams of data: dependability factors and the features defined by the type of business service realized - makes a starting point for practical tool for defining an organization of network systems maintenance.

It is possible to operate with large and complex networks described by various - not only classic – distributions and set of parameters. The model can be used as a source to create different measures – also for the economic quality of the network systems. The presented problem is practically essential for defining and organization of network services exploitation.

References

1. Al-Kuwaiti, M., Kyriakopoulos, N., Hussein, S.: A Comparative Analysis of Network Dependability Fault-tolerance, Reliability, Security, and Survivability. IEEE Communications Surveys & Tutorials 11(2), 106–124 (2009)
2. Arvidsson, J.: Taxonomy of the Computer Security Incident Related Terminology. Telia CERT (2006), `http://www.terena.nl/tech/projects/cert/i-taxonomy/archive/.txt` (retrieving date of access: May 15, 2011)
3. Avizienis, A., Laprie, J.C., Randell, B.: Fundamental Concepts of Dependability. Toulouse France: LAAS-CNRS Research Report No. 1145, LAAS-CNRS (2001)
4. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. Dependable and Secure Computing (TDSC) 1(1), 11–33 (2004)
5. Bonabeau, E.: Agent-Based Modelling: Methods and Techniques for Simulating Human Systems. Proc. Natl Acad. Sci. (2002)
6. Gao, Y., Freeh, V.W., Madey, G.R.: Conceptual Framework for Agent-based Modelling and Simulation. In: Proceedings of NAACSOS Conference, Pittsburgh (2003)
7. Jennings, N.R.: On Agent-Based Software Engineering. Artificial Intelligence 117, 277–296 (2000)
8. Kołowrocki, K.: Reliability of Large Systems. Elsevier, Amsterdam (2004)
9. Kyriakopoulos, N., Wilikens, M.: Dependability and Complexity: Exploring Ideas for Studying Open Systems, EN. EC Joint Research Centre, Brussels (2001)
10. Lapie, J.C.: Dependability: Basic Concepts and Terminology. Springer, New York (1992)
11. Liu, H., Chu, L., Recker, W.: Performance Evaluation of ITS Strategies Using Microscopic Simulation. In: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems, pp. 255–270 (2004)
12. Mascal, C.M., North, M.J.: Tutorial on Agent-Based Modelling and Simulation. In: Winter Simulation Conference (2005)
13. Walkowiak, T., Mazurkiewicz, J., Nowak, K.: Fuzzy Availability Analysis of Web Systems by Monte-Carlo Simulation. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 616–624. Springer, Heidelberg (2012)
14. Melhart, B., White, S.: Issues in Defining, Analyzing, Refining, and Specifying System Dependability Requirements. In: Proc. of the7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2000), April 3-7, pp. 334–340. IEEE Computer Society, Edinburgh (2000)
15. Mellouli, S., Moulin, B., Mineau, G.W.: Laying Down the Foundations of an Agent Modelling Methodology for Fault-Tolerant Multi-agent Systems. In: Omicini, A., Petta, P., Pitt, J. (eds.) ESAW 2003. LNCS (LNAI), vol. 3071, pp. 275–293. Springer, Heidelberg (2004)
16. Michalska, K., Mazurkiewicz, J.: Functional and Dependability Approach to Transport Services Using Modelling Language. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 180–190. Springer, Heidelberg (2011)

17. Michalska, K., Walkowiak, T.: Hierarchical Approach to Dependability Analysis of Information Systems by Modeling and Simulation. In: Cotton, A., et al. (eds.) Proceedings of the 2nd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2008), Cap Esterel, France, August 25-31, pp. 356–361. IEEE Computer Society Press, Los Alamitos (2008)
18. Michalska, K., Walkowiak, T.: Modelling and Simulation for Dependability Analysis of Information Systems. In: Świątek, J., et al. (eds.) Information Systems Architecture and Technology. Model Based Decisions, pp. 115–125. University of Technology, Wroclaw (2008)
19. Nowak, K.: Modelling of Computer Systems – an Approach for Functional and Dependability Analysis. In: Kołowrocki, K., Soszyńska-Budny, J. (eds.) Journal of Polish Safety and Reliability Association, Summer Safety and Reliability Seminars (SSARS 2011), vol. 1, pp. 153–161 (2011)
20. Walkowiak, T., Mazurkiewicz, J.: Availability of Discrete Transportation System Simulated by SSF Tool. In: Proceedings of International Conference on Dependability of Computer Systems, Szklarska Poreba, Poland, pp. 430–437. IEEE Computer Society Press, Los Alamitos (2008)
21. Walkowiak, T., Mazurkiewicz, J.: Functional Availability Analysis of Discrete Transportation System Realized by SSF Simulator. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 671–678. Springer, Heidelberg (2008)
22. Walkowiak, T., Mazurkiewicz, J.: Algorithmic Approach to Vehicle Dispatching in Discrete Transportation Systems. In: Sugier, J., et al. (eds.) Technical Approach to Dependability, pp. 173–188. Wroclaw University of Technology, Wroclaw (2010)
23. Walkowiak, T., Mazurkiewicz, J.: Functional Availability Analysis of Discrete Transportation System Simulated by SSF Tool. International Journal of Critical Computer-Based Systems 1(1-3), 255–266 (2010)
24. Walkowiak, T., Mazurkiewicz, J.: Soft Computing Approach to Discrete Transport System Management. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 675–682. Springer, Heidelberg (2010)
25. Volfson, I.E.: Reliability Criteria and the Synthesis of Communication Networks with its Accounting. J. Computer and Systems Sciences International 39(6), 951–967 (2000)
26. Xiaofeng, T., Changjun, J., Yaojun, H.: Applying SOA to Intelligent Transportation System. In: Proceedings of the IEEE International Conference on Services Computing, July 11-15, vol. 2, pp. 101–104. IEEE Computer Society, Orlando (2005)
27. Zamojski, W., Caban, D.: Assessment of the Impact of Software Failures on the Reliability of a Man Computer System. In: Proc. of the Conference on European Safety and Reliability (ESREL), pp. 2087–2090. A. A. Balkema, Gdynia-Sopot-Gdansk (2005)
28. Zhou, M., Kurapati, V.: Modelling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific Publishing, London (1999)
29. Zhu, J., Zhang, L.Z.: A Sandwich Model for Business Integration in BOA (Business Oriented Architecture). In: Proceedings of the IEEE Asia-Pacific Conference on Services Computing (APCSC), pp. 305–310. IEEE Computer Society, Washington, DC (2006)

A Comparison of Forecasting Methods for Ro-Ro Traffic: A Case Study in the Strait of Gibraltar

José Antonio Moscoso López[1], J.J. Ruiz-Aguilar[1], I. Turias[1], M. Cerbán[2], and M.J. Jiménez-Come[1]

[1] Intelligent Modelling of Systems Research Group, Polytechnic School of Engineering (Algeciras). University of Cádiz. Avda. Ramón Puyol s/n, 11202 Algeciras (Cádiz), Spain
{joseantonio.moscoso,juanjesus.ruiz, ignacio.turias,mariajesus.come}@uca.es

[2] Research Group Transport and Innovation Economic, Faculty of Economics. University of Cádiz. Avda. Duque de Nájera s/n, 11002 Cádiz, Spain
mariadelmar.cerban@uca.es4

Abstract. The objective of this article is to predict volumes of Ro-Ro (Roll-on, Roll-off) freight in order to apply this prediction as a decision making tool in logistics planning and port organization. This tool can help to improve supply chain performance in a Ro-Ro terminal. Seasonal ARIMA (SARIMA) and Artificial Neural Networks (ANNs) were the forecasting methods used in this study. A resampling procedure was applied in order to find out the best model from a statistical point of view using multiple comparison methods. The results have been very promising (*R=0.9157; d=0.9546; MSE=0.0195*)

Keywords: Forecasting, Logistics planning, Decision making tool.

1 Introduction

Management of the logistic and transport chain plays a key role in maritime and port environments. Ports have lost importance as nodes within the transport chain due to the fact that major logistic operators have become critical actors.

Freight transport is a relevant economic issue in most countries. In this sense, the efficiency of the transport system is an essential influence on the location of the companies [1]. The improvement of the transport systems is closely linked to the availability of advanced planning tools [2-3]. The use of these tools makes it possible to analyse current trends and to make predictions. Other studies have stated that the prediction of demand is critical to design a logistic and transport chain according to the current geographical necessities [4].

Short-term predictions have been used to facilitate the implementation of daily port operations activities, such as the allocation and provision of personnel and the necessary equipment [5].

The planning process is very complex, difficult to model, and is performed in a dynamic environment which implies that traditional prediction tools are not effective.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_33,

A viable alternative would be the combination of different methodologies, with ANNs or SARIMA outstanding as a possibility [6-7].

In general, terms, modelling is intended to provide a tool for transport planners and Port Authorities. This tool improves supply chain performance in a Ro-Ro terminal [8] and could be useful to the different actors involved in port logistics. Several authors have been made reliable traffic predictions using different approaches such as ARIMA, linear regression or ANNs [5], [9-10].

ARIMA models have been used in different studies on traffic predictions. Godfrey & Powell [11] performed a daily demand prediction freight transport using various methods, including ARIMA. ARIMA models were also used to compare with other methods such as BPNN [12]. On the other hand, Chung & Rosalion [13] proposed a short term forecasting on Melbourne with some methods including ARIMA and SARIMA.

ANNs are used in different areas of transport research. There are studies on traffic flow predictions for Intelligent Traffic Systems (ITS) [14-16]. Cantarella and de Luca [17] performed a simulation of traffic demands and choice of transport mode in two different areas, using Multilayer Perceptron networks (MLP), while [18] used Back Propagation Neural Network (BPNN) in order to predict and to model public transport trip demand of commercial sites.

The aim of this paper is to design the best model for predicting future values of Ro-Ro traffic at the Port of Algeciras Bay in the Strait of Gibraltar. These predictions can be used as a decision making tool in logistics planning and organization (the prediction horizon is one day). The freight chosen for this study is vegetables due to the fact that they represent the largest category of Ro-Ro traffic in the Strait of Gibraltar and accounted for 32% of the total freight.

The next section presents the methodology used in this work; Section 3 analyses the main results, and Section 4 establishes the conclusions.

2 Methodology

The real problem is to estimate the future value of a time series according to its value in the present and previous values in a certain window of *n*-samples in the past. Thus, the estimate can be modelled by a function, generally nonlinear, of the previous values of the time series (1).

$$\hat{y}(t+1) = f(y(t)...y(t-n)) \tag{1}$$

2.1 Seasonal Autoregressive Integrated Moving Average (SARIMA)

The ARIMA approach was first popularized by Box and Jenkins [19]. The general notation for the order of an ARIMA model is ARIMA*(p, d, q)*, where p is the number of autoregressive terms *(AR)*, *d* is the number of non-seasonal differences *(I)* and *q* is the number of lagged forecast errors in the prediction equation *(MA)*. If the time series

is seasonal, a seasonal ARIMA (SARIMA) should be applied to handle the seasonal aspects of the time series and the general notation is ARIMA$(p,d,q)(P,D,Q)_s$.

The equation for the ARIMA$(p,d,q)(P,D,Q)_s$ models is (2).

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)e_t \tag{2}$$

Where

$\Phi_1 \ldots \Phi_p$: are the seasonal autoregressive coefficients.

$\Theta_1 \ldots \Theta_2$: are the seasonal moving average coefficients.

Y_t: Variable to predict.

B: is the backshift operator.

S: is the period of study in days.

e_t: is the error term.

2.2 Artificial Neural Networks

In this paper, a multilayer perceptron (MLP) with feedforward connections has been used since this type of neural network is the most widely used model in the literature. Multilayer feedforward networks are known to be universal function approximators [20]. Backpropagation learning procedure [21] has been used in this work. In BPNN, the weights are adjusted to minimize the mean square error between the desired output and the actual network output. It is not necessary to know the error function analytically, instead an iterative process of minimizing a measure of the error made by the model, based on the available samples, can be used. In the classic algorithm, a gradient descent method is used, but there are many other algorithms for training multilayer neural networks: conjugate gradient, quasi-Newton, Levenberg-Marquardt, etc. In this paper, the Levenberg-Marquardt algorithm has been used, which operates at first as a gradient descent method and then as a quasi-Newton method with approximation of the Hessian. This algorithm provides sufficient robustness and velocity [22].

This neural approach models the relationship between X and Y in the form (through a weighted structure of layers, usually input-hidden(s)-output),

$$Y = g\left(\sum_{j=0}^{M} w_{kj} \cdot f\left(\sum_{i=0}^{D} w_{ij} \cdot X_i\right)\right) \tag{3}$$

Where W_{ij} is the matrix of the connection weight input layer with D being the total number of the inputs units and W_{kj} the matrix of the connection weight hidden layer with M number of units (3).

2.3 Experimental Procedure

It was necessary to perform an experimental design to determine the best model (i.e., the one with the lower generalization error). The generalization error must be measured in order to determine how the model will behave with real patterns (not with training patterns).

The experimental procedure to obtain the best SARIMA model consists of performing an iterative procedure by choosing different configurations *(p,d,q)(P,D,Q)s*. Table 1 shows the values of the parameters of SARIMA*(p,d,q)(P,D,Q)s* that were combined (in total 192 models). In order to check the performance of each model, the database was divided into two data sets: data from January 2000 to December 2006 were used to estimate the model and those from January 2007 to December 2007 were used to test the model.

Table 1. Values of SARIMA$(p,d,q)(P,D,Q)_s$ parameters

Non-seasonal Parameters	Non-Seasonal values	Seasonal Parameter	Seasonal Values
p	0-3	P	0-2
d	1	D	1
q	0-3	Q	0-3
		s	7

The experimental procedure to obtain the best ANN model consists of applying a multiple comparison scheme: Different ANN models based on multilayer perceptron MLP with backpropagation learning algorithm have been tested. Levenberg-Marquardt has been used as optimization algorithm. A two-fold crossvalidation procedure has been used in order to measure the generalization error using the correlation coefficient (*R*), the index of agreement (*d*) and the mean square error (*MSE*), for test sets. In order to select the best model it is necessary to test several network configurations with different numbers of hidden units (nhiddens) as well as different values of the autoregressive window size (*n*).

Those different ANN models are shown in Table 2.

Table 2. ANN Models tested in the experiment for each n

Autoregressive window size *n* (1, 2, 7, 14, 21, 28)		
Model number	*nhiddens*	*epochs*
(1-35)	(1, 2, 5, 10, 15, 20, 30)	(100, 300, 500, 700, 900)
1-5	1	100,300, …, 900
6-10	2	100,300, …, 900
⋮	⋮	⋮
31-35	30	100,300, …, 900

The multiple comparison procedure was applied in a two-stage scheme [23]. In Phase-I, Friedman test aimed to test the null hypothesis with a significance level $\alpha=0.05$ for each different size *n*. When the null hypothesis was rejected, a LSD Fisher post hoc test was applied. In Phase-II, the same process was repeated with all models obtained in Phase-I. These models were compared and the best overall model was determined. This procedure, shown in Fig. 1, was applied to each performance index (*R*, *d* and *MSE*).

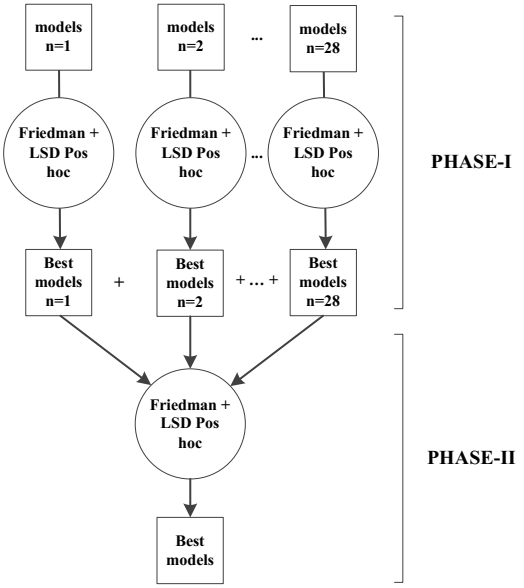


Fig. 1. Two-stage multiple comparison procedure to obtain the best model with ANNs for R, d and MSE independently

3 Results

The database comprises all trucks that carry vegetable freight and cross the Strait of Gibraltar from 1 January 2000 to 31 December 2007 (2970 days). The tonnes of this freight have increased by 167% in the study period (Fig. 2a). On the other hand, the 79% has crossed the Strait of Gibraltar between November and April (Fig. 2b).

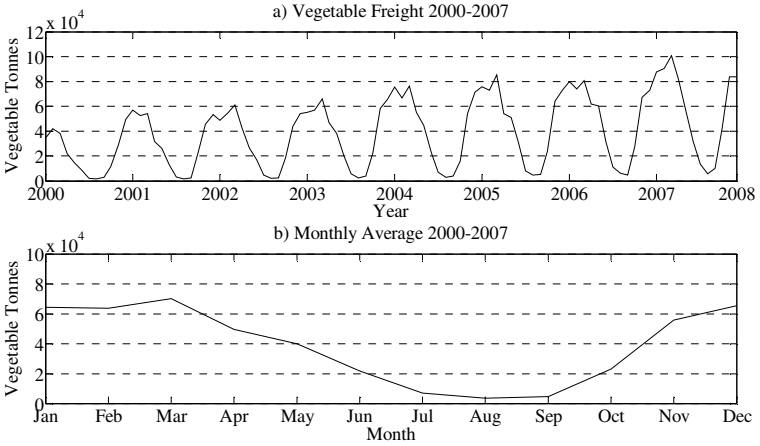


Fig. 2. Tonnes of vegetable Ro-Ro freight in the Strait of Gibraltar (2000-2007); a) Total Tonnes. b) Monthly average tones.

Table 3. Best results of the iterative procedure for SARIMA models

Model	*R*	*d*	*MSE*
SARIMA(1,1,1)(1,1,1)$_7$	0.9098	0.9530	0.2367
SARIMA(1,1,1)(1,1,2)$_7$	0.9096	0.9529	0.2372
SARIMA(1,1,2)(1,1,2)$_7$	0.9105	0.9533	0.2337
SARIMA(2,1,1)(2,1,1)$_7$	0.9121	0.9535	0.2257
SARIMA(1,1,1)(2,1,1)$_7$	0.9125	0.9540	0.2255
SARIMA(1,1,1)(2,1,2)$_7$	**0.9125**	**0.9540**	**0.2254**
SARIMA(1,1,2)(2,1,1)$_7$	0.9120	0.9535	0.2259
SARIMA(2,1,1)(2,1,2)$_7$	0.9118	0.9531	0.2275
SARIMA(2,1,1)(2,1,1)$_7$	0.9121	0.9535	0.2257
SARIMA(3,1,3)(2,1,1)$_7$	0.9116	0.9532	0.2266

SARIMA models were evaluated with different configurations $(p,d,q)(P,D,Q)_s$, following the experimental procedure explained above.

Table 3 compares the results achieved from the best SARIMA models configurations. SARIMA$(1,1,1)(2,1,2)_7$ model obtained the smallest *MSE*. It is marked in bold in Table 3.

In the case of ANN models, mean values of *R*, *d* and *MSE* performance indexes are represented in Fig. 3. The best results are pointed out with black colour for *R* and *d* and with grey for *MSE*. It is worth mentioning that when the number of *nhiddens* increases, the values of the indexes get worse. After the two stages of the multicomparison process, twenty four not significantly different models were obtained as the

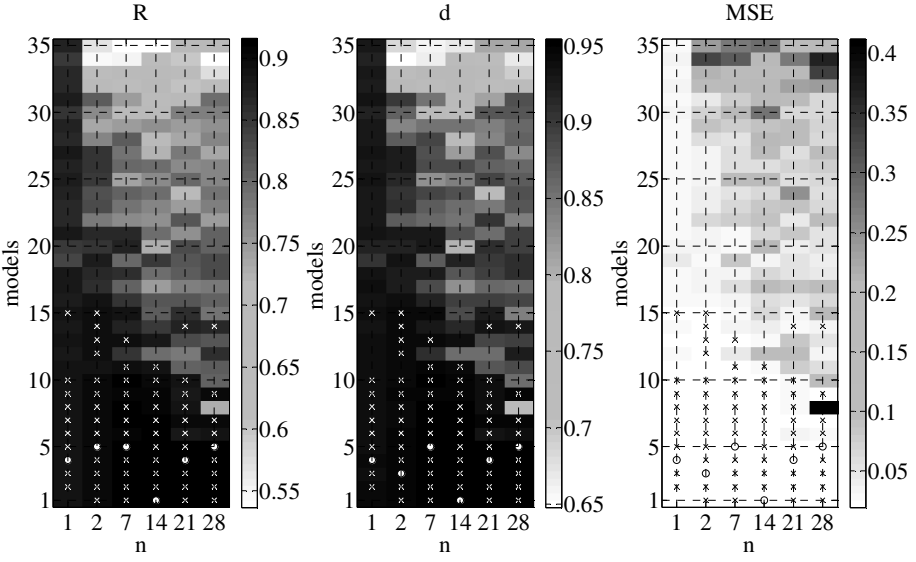


Fig. 3. Multicomparison procedure results: *R*, *d* and *MSE* values for the different sizes of autoregressive windows (*n*). Crosses are models without differences and circles are the best models for each *n*.

best models of all those analyzed. Table 4 shows the results obtained from Phase-II pointed out that the best ANN model was the one with *n=14*, one hidden neuron layer. Therefore, the use of a larger auto-regressive window of fourteen days does not provide better results.

The results of the prediction (for test set) of the best ANN model in one of the 30 repetitions of the experimental procedure is represented in Fig. 4.

Table 4. Best models (not significantly different) obtained from Phase-II and results of *R*, *d* and *MSE*. The best model is shown in bold

n	*nhidden*	*epochs*	*R*	*d*	*MSE*	*n*	*nhidden*	*epochs*	*R*	*d*	*MSE*
7	1	100	0.9142	0.9537	0.0198	14	2	100	0.9108	0.9522	0.0206
7	1	300	0.9140	0.9535	0.0198	14	2	300	0.9111	0.9524	0.0206
7	1	500	0.9138	0.9535	0.0199	21	1	100	0.9152	0.9543	0.0197
7	1	700	0.9140	0.9536	0.0198	21	1	300	0.9149	0.9542	0.0197
7	1	900	0.9147	0.9538	0.0197	21	1	500	0.9151	0.9542	0.0197
7	2	300	0.9135	0.9533	0.0200	21	1	700	0.9154	0.9544	0.0196
7	2	900	0.9120	0.9527	0.0203	21	1	900	0.9151	0.9543	0.0197
14	1	100	0.9157	0.9546	0.0195	28	1	100	0.9147	0.9542	0.0198
14	1	300	0.9154	0.9544	0.0195	28	1	300	0.9151	0.9543	0.0197
14	1	500	0.9151	0.9543	0.0196	28	1	500	0.9145	0.9540	0.0199
14	1	700	0.9149	0.9541	0.0197	28	1	700	0.9142	0.9538	0.0199
14	1	900	0.9150	0.9542	0.0197	28	1	900	0.9152	0.9545	0.0197

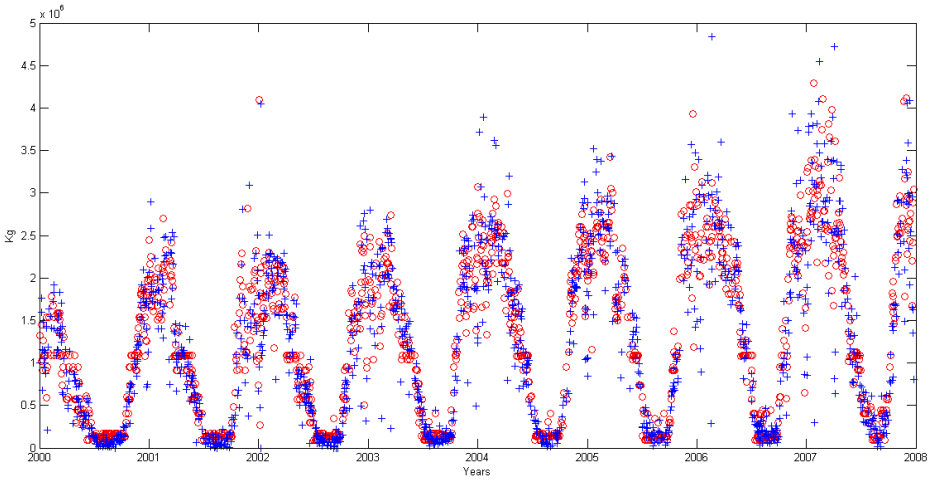


Fig. 4. Volume prediction of vegetable Ro-Ro freight in the 2000-2008 period. Circles are the predictions of the best model (ANN with *n=14* and *nhiddens=1*) for the test set. Crosses are the real values

The results of the best SARIMA and ANN models are compared in **Table 5**. The values of *R* and *d* are very similar in both models however *MSE* reveals that the ANN model outperformed the SARIMA model. The best values are shown in bold. Therefore, ANNs are more suitable (pointing out its better *MSE* values) for forecasting this time series of vegetables in the Port of Algeciras Bay with excellent results.

Table 5. Comparison of the best models ANNs and SARIMA

Model	*R*	*d*	*MSE*
SARIMA$(1,1,1,)(2,1,2)_7$	0.9125	0.9540	0.2254
ANN (n: 14; nhiddens: 1; epochs: 100)	**0.9157**	**0.9546**	**0.0195**

4 Conclusions

The results showed that in this case ANNs provide excellent forecast results, improving the results of SARIMA. This study used a random resampling procedure, which allows the statistical comparison of models by generalization error, measured using specific quality indexes for test sets. ANN models are capable of acquiring the seasonality of the time series implicitly. These results provide a tool to predict the level of Ro-Ro traffic with excellent accuracy (*R=0.9157; d=0.9546; MSE=0.0195*), a fact that may be helpful in decision making for different port agents, e.g. in the preparing for port infrastructure or the assignment of different workloads.

Finally, this analysis has stressed the usefulness of ANNs for improving performance and competitiveness of the transport and logistics chain.

Acknowledgements. We thank the Port Authority of the Port of Algeciras Bay for providing the database. In addition, this work is supported in part by a grant from the European Project FEDER-FSE 2007-2013 and the Technological Campus Foundation of Algeciras Bay.

References

1. Apivatanagul, P., Regan, A.C.: Long haul freight network design using shipper–carrier freight flow prediction: A California network improvement case study. Transportation Research Part E: Logistics and Transportation Review 46, 507–519 (2010)
2. Walkowiak, T., Mazurkiewicz, J.: Analysis of critical situations in discrete transport systems. In: Fourth International Conference on Dependability of Computer Systems, DepCos-RELCOMEX 2009, p. 364. IEEE (2009)
3. Bianco, L., La Bella, A.: Freight transport planning and logistics: Lucio Bianco and Agostino La Bella. Springer, New York (1988)
4. Hesse, M., Rodrigue, J.: The transport geography of logistics and freight distribution. Journal of Transport Geography 12, 171–184 (2004)
5. Peng, W.Y., Chu, C.W.: A comparison of univariate methods for forecasting container throughput volumes. Mathematical and Computer Modelling 50, 1045–1057 (2009)

6. Dougherty, M.: A review of neural networks applied to transport. Transportation Research Part C: Emerging Technologies 3, 247–260 (1995)
7. Schulze, P.M., Prinz, A.: Forecasting container transshipment in Germany. Applied Economics 41, 2809–2815 (2009)
8. Dias, J.C.Q., Calado, J.M.F., Mendonça, M.C.: The role of European «ro-ro» port terminals in the automotive supply chain management. Journal of Transport Geography 18, 116–124 (2010)
9. Al-Deek, H.M.: Use of vessel freight data to forecast heavy truck movements at seaports. Transportation Research Record: Journal of the Transportation Research Board 1804, 217–224 (2002)
10. Mostafa, M.M.: Forecasting the Suez Canal traffic: a neural network analysis. Maritime Policy & Management 31, 139–156 (2004)
11. Godfrey, G.A., Powell, W.B.: Adaptive estimation of daily demands with complex calendar effects for freight transportation. Transportation Research Part B: Methodological 34, 451–469 (2000)
12. Van Der Voort, M., Dougherty, M., Watson, S.: Combining Kohonen maps with ARIMA time series models to forecast traffic flow. Transportation Research Part C: Emerging Technologies 4, 307–318 (1996)
13. Chung, E., Rosalion, N.: Short term traffic flow prediction. In: 24th Australasian Transport Research Forum (ATRF), Hobart, Tasmania, Australia (2001)
14. Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G.: Short-term traffic forecasting: Overview of objectives and methods. Transport Reviews 24, 533–557 (2004)
15. Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C.: Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. Transportation Research Part C: Emerging Technologies 13, 211–234 (2005)
16. Lan, J., Guo, M., Lu, H., Xiao, X.: Short-Term Traffic Flow Combination Forecast by Cointegration Theory. Journal of Transportation Systems Engineering and Information Technology 11, 71–75 (2011)
17. Cantarella, G.E., de Luca, S.: Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models. Transportation Research Part C: Emerging Technologies 13, 121–155 (2005)
18. Zhu, Y., Chen, Y., Geng, X., Liu, L.: Transport Modal Split of Commercial Sites Based on Artificial Neural Network. Journal of Transportation Systems Engineering and Information Technology 8, 86–91 (2008)
19. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, Oakland (1976)
20. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366 (1989)
21. Rummelhart, D.E., McClelland, J.L.: PDP Research Group, Parallel Distributed Processing. Explorations in the Microstructure of Cognition, vol. 1, Foundations (1986)
22. Hagan, M.T., Menhaj, M.B.: Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks 5, 989–993 (1994)
23. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7, 1–30 (2006)

Partial Blur: Model, Detection, Deblurring

Dmytro Peleshko, Mariya Rashkevych , Andriy Klyuvak, and Yuriy Ivanov

Lviv Polytechnic National University, Department of Information Technologies in Publishing
rashkev@polynet.lviv.ua

Abstract. Physical process of blurring emergence has been analyzed. It has been proved through conducted experiments that image blurring formation is adequately described by the model based on convolution, i. e. wrapping. It is shown that blurring center or discrete function of point scattering comprises information about trajectory and uniformity of motion, which has caused an image distortion. It determined that extreme values number of averaged normalized column values of Fourier image is distorted by artificial blurring correlates with parameters of blurring.

Keywords: blur, deconvolution, point spread function (PSF), distortion area, image reconstruction.

1 Introduction

Blurring is such a distortion type that emerges as a result of dynamic changes of attention objects or background during frame exposure.

While researching the image, blurred artificially or in a natural way, the biggest attention should be paid to the spectral image features. This determines construction and use of metrics, obtained on the base of the spectral image-signal features. Such an approach allows better understanding of nature and mechanism of natural blur formation, enabling the development of reconstructive technologies.

Formalization of the image distortion process as a result of motion comes to the construction of mathematical blur model that in its turn similar to the distortion case (result of refocusing) is determined by the operation of wrapping (convolution). Formal determination of the operation is the following. Some discrete distorted image $f(x, y)$ of the dimension $M \times N$ with the blur center $h(x, y)$ and its dimension $m \times n$ is given This center is titled Point Spread Function (PSF) [4]. In addition, light sensitive matrix imposes randomly distributive additive noise $n(x, y)$. Then resulting image $g(x, y)$ with the distortion, which emerged as a result of a camera motion, will be determined by the following formula:

$$g(x, y) = h(x, y) \otimes f(x, y) + n(x, y), \quad (1)$$

where the sign $\otimes$ means the wrapping operation.

The noise n is an integral part of up-to-date matrices, what is revealed via random deviations of intensity function values of the image point from real color value.

Noise reasons in digital sensors can be different. But in most cases it is Gaussian,

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_34,

which parameters are average value and dispersion that characterize the most frequent value and other value deviations. Noise is additive and does not correlate with the image and does not depend on the pixel location. This stochasticity complicates the procedures of reconstructive algorithms development for images, as far as despite of the distortion type, noise to a greater or lesser extent is always present.

The image distortion is determined by the value of PSF $h(x, y)$. The function determines the distortion type of every image pixel. During the process of distortion every pixel of an initial image changes into a spot in case of refocusing and in case of a certain blur into a segment. The same process can be represented in the other way: the intensity function value of every distorted image pixel is an integral characteristic of pixel values in not distorted image. A distorted image is formed as a result of these representations. The law, by which every pixel is distorted, is determined by the PSF.

In a discrete case the PSF can be represented by the matrix operator. As a rule, its dimension is less than the dimension of a whole image and is an important distortion characteristic. The intensity function value depends on the PSF, which determines the wrapping (convolution) operation and in (1) is depicted as the symbol⊗. Hereby some part of the initial image is wrapped into one pixel.

From the wrapping operation it follows that the sum of all PSF matrix values is equal to one. Physical content of the PSF matrix values lies in such definition when every of them stands for the light proportion of the given point, which is observed in the other coordinates. Obviously, in any convolutional image distortion the intensity function value of every point is distributed over different coordinates without any remainder.

There are typical distortion functions. If in case of refocusing of an image, every point is transformed into a spot, then all not null elements of the PSF matrix are located in the form of spherical spot with a radius of the every point dispersion.

In case of motion distortion, i.e. blurring, all not null elements of the mentioned matrix are located along the trajectory of distortion motion. This is predetermined by the fact that every point is blurred along the trajectory during exposure.

2 Frequency Characteristic of Blurring

Frequency characteristics are the most interesting for the research of blurred images, as far as they reveal the character and parameters of distortion. To research the frequency characteristics of signals the Fourier transform (FT) is used [1]. In accordance with [1, 2, 6] we can present the final spectrum of the image distorted by the camera motion via the discrete FT.

The blurring model adequacy verification on the basis of wrapping (convolution) operation should be conducted with the help of two types of images: the images distorted in a natural way, i.e. by the camera motion, and images artificially distorted by the wrapping operation.

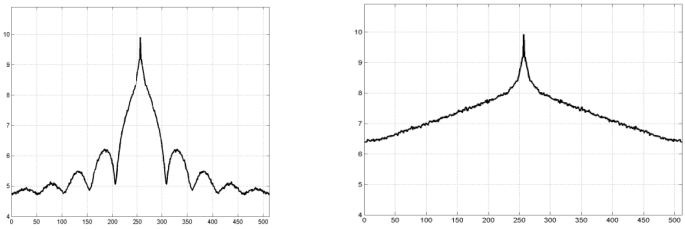
First, the convolution operation with the known blurring center is conducted. The blurring center is chosen to imitate the camera motion. This means that in case of a full blurring along the image plane the blurring center will represent one of the

simplest distorting motions. If the motion is uniform and strictly horizontal with the length of 10 points, then the blurring center will be a vector-row with the dimension of 10 elements; value of every element will be 0,1. Such an approach to the blurring center construction is accepted only because of the implementation simplicity, as far as the motion character is not principal while determining the blurring nature.

Comparative results are presented in Fig. 1. The right part of Fig. 1 refers to the output image, and the left – to the result of artificial distortion by the horizontal motion center. All its constituents, both real and imaginary parts have chaotic character similar to the noise, what is characteristic for natural images.

Instead some regularity in the distorted image, which is shown in vertical periodical lines, appears. These lines are present in both real and imaginary parts of FT of images (Fig. 2a, [3]).

Fig. 1. Vizualization of discrete FT results before and after artificial distortion by horizontal blurring center



a) b)

Fig. 2. Averaged normalized quantities of column values of FT of: a) the image, distorted by the artificial horizontal blur b)natural not-distorted image.

To compare the same characteristic but for the case of not distorted image (the right part of Fig. 1) consider the Fig. 5b. The main difference between graphs in Fig. 2a,b lies in the presence of the distorted image of periodical extreme values in the graph. A natural, not distorted image, has no such extreme values.

Statement 1.1 – The number of extreme values of averaged normalized values of FT columns of an image that is distorted with an artificial blur correlates with the parameters of this blur.

To confirm the statement 1.1 let's consider numeral parameters of the researched image and artificial horizontal blur. The results represented in the Fig. 2a concern the artificial horizontal blur of 10 pixels. The number of extreme values is equal to 9. The central extreme value is of doubled thickness, what is equal to the sum of 10 e

extreme values. The regularity can be explained by repetitions of the intensity function values by imposing color onto the neighboring points while blurring. That is why the number of extreme values is very close to the blur length. In the ideal case they are equal because every point is imposed onto the other ten points.

The similar experiment has been repeated with other 20 images. The case of vertical blur was considered separately. The results of all experiments were similar to those described.

Statement 1.2 – The number and frequency of extreme values of averaged normalized values of FT image, distorted by an artificial blur, is an invariant for the change of the blur parameters.

The formulated statement determines that there are typical blurring parameters, which are independent from the change of blurring operator parameters. Several more experiments with the same initial image but different blur parameters have been conducted to verify the conclusion. Fig. 3 and 4 illustrate the results of the experiments for the case of artificial horizontal blur. The given graphs of experimental research confirm the correctness of statement on invariance.

The final verification of the blur model adequacy lies in comparison of the data obtained during studies of the artificially distorted images with the corresponding data for images with natural blurring. All of the studies described above and their results with the artificially distorted images were performed with the natural image and with the clear blurring shown in the Fig. 5. The results of these studies, in particular a frequency graph and its images obtained like in Fig. 3, 4, are shown in Fig. 5.

The results suggest that natural images contain much more noise than that blur which was obtained artificially. This is proved by a greater number of small extreme values on the graph of Fig. 5. Obviously, a natural image pattern is not as clear and unambiguous as the image with the artificial blur. However, it is possible to separate features specific for the horizontal blur case. With the pixel by pixel analysis of the image it has been determined that the blur length is 20 pixels. The 18 regular extreme values can be separated, what is sufficiently close to the real blur parameter as it is shown on the graph in Fig. 5. These peaks are similarly visible on the synthesized image shown in Fig. 5.

Taking into consideration the fact that in cases of natural and artificial distortions the identical spectral characteristics are proved, we can conclude that the representation of the blur model based on the convolution operation is adequate. Hence, the reconstructive algorithms based on the deconvolution operation can be implemented to eliminate blur on natural images.

Despite the existence of few sufficiently effective algorithms of reconstruction of images distorted by blurring, there is no method to restore such an image with no lost features of informational content. The value of such losses is a measurement of effectiveness of the reconstruction algorithm.

To understand the reason of such losses, first of all it is necessary to check the correctness of chosen model of image blurring. Typically this verification should be completed via procedure of blurring simulation on non-distorted image. Aiming at this, the image is artificially distorted by some motion based on previously known parameters (trajectory and uniformity). Further, according to previously determined metrics we make a comparison with the image that has been distorted in a natural way while exposure of a movable object by the device with matrix sensitive to the light.

Fig. 3. Vizualization of discrete FT results for the image with artificial horizontal blur of 5 pixels

Fig. 4. Vizualization of discrete FT results for the image with artificial horizontal blur of 20 pixels

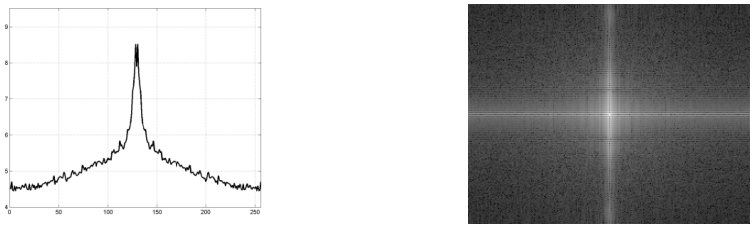


Fig. 5. Vizualization of discrete FT results for the image with natural blur

Methods of distorted image reconstruction, which are based on deconvolution transformation, are one of the most used in practice [8].

We differentiate algorithms of both blind and not blind convolution among deconvolution algorithms [7, 8]. A part of methods have a priori nature, and some of them anticipate the use of special hardware. Despite the type, the use of blurring center, i.e. PSF, for all algorithms is characteristic for all algorithms. The blurring center is PSF that caused blurring (distortion) of an image.

Signal distortion under influence of some blurring center (core) means that each distorted image point is a result of wrapping operation. The wrapping operation can be presented as follows:

$$d_i = \sum_j h_{i.j} c_j \ , \tag{2}$$

where $h_{i,j}$ – discrete PSF is an array (matrix), every element of which determines proportion of light from the point j, which is present in the point i; d – value of the intensity function in the point i after wrapping operation; c – intensity function value in the point i, i.e. such value that existed before the wrapping operation. As a rule, image distortion, caused by wrapping, is eliminated by one of deconvolution methods.

3 Model of Local Image Distortion Formation

Similar to distortion from defocusing of optical system, blurring nature lies in the following: information about the color of every point is distributed all over the image by some law. The difference between different types of distortion lies in the law, by which the distribution occurs. The law is determined by PSF.

To simplify the representation of local distortion formation model, let's consider one dimensional case, i.e. the action of some PSF on a certain vector $\{c_i | \ i = 1..4\}$, which elements are pixel values of a set image.

In case of any blurring, the intensity function value at the point of blurring area obtains color intensity. In particular, in case of horizontal blurring per a pixel as a result of distortion every pixel values and a value of a preceding (left) pixel by the coordinate are added and divided by two: $c'_i = (c_i + c_{i-1})/2$. This formula result from the following: as far as the left pixel comes on the given one during its movement, during exposure both values managed to reflect in the position. As a result we obtain a new distorted image:

$$(c_1 + c_0) / 2 \mid (c_2 + c_1) / 2 \mid (c_3 + c_2) / 2 \mid (c_4 + c_3) / 2. \quad (3)$$

This is the model of ideal distortion. The blurring analysis above dealt with a general case with no regard of its type. In accordance with the classification there are global (full) and local (partial) blurs. The final is more complicated than the first. That is why let's dwell upon the formation model of the local blurring, which covers not the whole image but some its part. Hereby, the rest is not distorted. On the contrary to the full blurring that is formed as a result of movement of a camera on immovable background, the local blurring has different mechanisms of formation. In compliance with this it is necessary to consider different models of distortion.

There are several types of local changes. One of them is formed when a camera, which exposes a frame with the moving object in it, is fixed.

The scheme of formation of such blurred area is the following. At the moment t_n shutter closes. During this period the moving object has moved to some distance, which taking into account discrete nature of a digital photo can be estimated by a finite number of pixels m. This makes possible to divide the time period $\Delta t = t_n - t_0$ into m equal interims. During every interim each pixel of the movable object left imprint of its value of intensity function at the other point via overlapping of the own intensity function value and the previous value at the same point. As a result values of intensity function in inner object pixels collide on the values of other pixels of the same object and the classic deconvolution problem emerges.

At the end of blurring section the situation is different. The ends can be considered as the area along the perimeter of an object in the movement direction and with the width of m points (buffer area). In the area the mix of values of intensity function of the moving object and values of intensity function of immovable background pixels occurs. As far as the exposure time was sampled into m periods, it can be considered that during time unit $\tau = 1/m$ intensity function value of every point of the buffer area is formed with τ part of the color value of the moving object and $(1-\tau)$ part of the value rest, which in its turn is formed as a result of additive overlapping in the same proportion of intensity function values, which belong to the object and background, which the object is moving over. For the limit object pixel, any intensity function

value will be determined by the described correlation. For the next object pixel its intensity function value $c(x_i, y_i)$ will be calculated by the scalar product of vectors $\mathbf{v} = (\tau, \tau, (1\text{-}2\tau))$ i $\mathbf{F}_i = (f(x_i, y_i), f(x_{i\text{-}1}, y_i), f_\Phi(x_i, y_i))$:

$$c(x_i, y_i) = \mathbf{v} \cdot \mathbf{F}_i. \tag{4}$$

Here $f_\Phi(x_i, y_i), f(x_i, y_i), f(x_{i\text{-}1}, y_i)$ – values of intensity function in the given and limit pixels. By the similar scheme intensity function values will be determined in every buffer area point.

The described above approach of formation of intensity function values in buffer area pixels will be named as the operation of weighing intensity function values, and the vector $\mathbf{v}$ – weighing operator. It should be noted that the indicated above parameters of the operator v are possible only under the condition of uniform motion. In case of nonuniform motion these parameters would be different and formed by the rule: the less period of object pixel stay at the position, the less is its part of intensity function value in the resulting value. However, the regularity obtained for the case of uniform motion will be preserved for the j pixel of the buffer area and can be written:

$$c_j = a_j f_j \ + (1\text{-}\ a_j) b_j \,, \quad a_j = \sum_{i=1}^{j} h_i \,, \tag{5}$$

where $j \in [0; m]$; h_i – i not null value of discrete PSF h; b_j – value of intensity function of background at the given point; f_j – integrated value of intensity function in pixels of any moving object, which were at the given position during the object motion.

By the other words, the configuration and dimension of the matrix PSF depend upon velocity, uniformity, motion trajectory, and frame exposure time. Arrangement of not null elements of the PSF matrix repeats the object motion trajectory for the period while the light-sensitive matrix exposed the given frame. The matrix elements values are proportional to the object velocity during the period from t_n to t_{n+1}, which is equal to $1/m$ of frame exposure time, where m – a number of not null matrix elements.

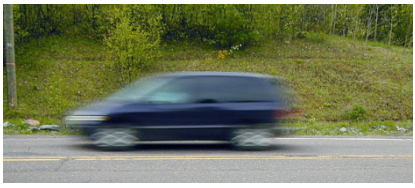


Fig. 6. Drawn nearer separation of buffer areas of partial distortion by the motion from a previous image

Fig. 7. Natural image, partially distorted by motion

Appropriately, during uniform motion not null matrix elements will be almost equal, and during linear motion they will be built in a line. A row vector results from the strict horizontal motion, and a column vector – from the strict vertical motion.

Hereby, it can be concluded that the buffer area under research repeats the PSF configuration: in case, if PSF is a vector-column of five elements, then the buffer area width will be 5 points of the moving object image.

Proper example that fully illustrates the idea of buffer zone can be taken from the following natural image (Fig. 6). Drawn nearer separation of both buffer areas is illustrated in Fig. 7.

4 Conclusions

Physical process of local blurring emergence and detailed mechanism of formation of distorted image by partial motion have been analyzed. Mechanism of buffer area formation between moving object of forefront and immovable background, i.e. transition area between an area of full blurring and non-distorted area, has been studied.

It has been shown that the blur center or discrete PSF comprises information about trajectory and motion uniformity, which caused the image distortion.

It has been found that the difference between the Fourier images distorted by the full image blur and not-distorted images is equal to the sum of exponential functions. Based on this formulated statement, the extreme values number of the averaged normalized values of the Fourier transform columns (in case an image is distorted by the artificial blur) correlates with the blur parameters.

It has been proved that the extreme values number and frequency of averaged and normalized values of the Fourier transform of the image distorted by the artificial blur are invariant to the changes of the blur parameters.

Proposed methodology of buffer area determination could be the base for development of effective deconvolution methods for elimination of global and local distortions, which emerge as a result of movement of an object or registration device.

References

1. Schuon, S.: The Nature of Motion Blur, http://ai.stanford.edu/~schuon/deblur/download/schuon_nature_of_motion_blur.pdf
2. Oliveira, J.P., Figueiredo, M.A.T., Bioucas-Dias, J.M.: Blind estimation of motion blur parameters for image deconvolution. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4478, pp. 604–611. Springer, Heidelberg (2007)
3. Molina, R., Mateos, J., Katsaggelos, A.K.: Blind deconvolution using a variational approach to parameter, image, and blur estimation. IEEE Transactions on Image Processing 15(12), 3715–3727 (2006)
4. Lucy, B.: An iterative technique for the rectification of observed distributions. Astronomical Journal 79(6), 745–754 (1974)
5. Vaseghi, S.V.: Advanced Digital Signal Processing and Noise Reduction, 3rd edn., p. 453. John Wiley & Sons Ltd. (2006)
6. Бузовский О.В. Компьютерная обработка изображений/ О. В. Бузовский. – Киев: Корнийчук, 79с (2001)
7. Васьков С. Т. Быстрая цифровая реконструкция сигналов и изображений по критерию минимума энергии/ С. Т. Васьков, В. М. Ефимов, А. Л. Резник// Автометрия 39(4), 14–20 (2003)
8. ЮжиковВ.BlindDeconvolution — автоматическое восстановление смазанныхзображений. Режим доступу, http://habrahabr.ru/post/175717/

Software Support for Common Criteria Security Development Process on the Example of a Data Diode

Dariusz Rogowski

Institute of Innovative Technologies EMAG, 40-189 Katowice, Leopolda 31, Poland
drogowski@emag.pl

Abstract. The data diodes are very often used to protect users' networks and sensitive data and that is why additional assurance of those devices is demanded. This assurance can be obtained by applying the Common Criteria security development process. The process is very difficult and time-consuming specially for those not familiar with the standard. Although there are many guidelines and templates telling how to define the security problem still there is a lack of computer aiding tools. This paper describes the plug-in application which supports identification of protected assets, threats, security objectives and security functions – the main elements of security specification. The tool facilitates and speeds up the security development process of IT products.

Keywords: Common Criteria, security development, a data diode.

1 Introduction

Security of networked systems has received much attention in recent years due to many cyber attacks and leaks of sensitive information. There are devices and systems on the market which can help to protect higher security classified networks. For instance, it could be both software and hardware solutions such as firewalls or smart switches. Among them there is one very interesting and simple-structured hardware device that can prevent data leakage of confidential or classified data. This is the so called data diode which makes one-way link between a high security classified network and a low security classified network [1-5].

The data diode is built of fiber optic transceivers which enable one way data transmission. Fiber optic technology is used to minimize the electromagnetic radiation of a device by which transmitted data could be eavesdropped. The data diode connection ensures that unwanted access to the protected network is not possible while there is still free access to data sources placed outside. For instance, in the protected zone, anti-virus applications, operating systems, emails can be updated freely by downloading necessary data from public networks (e.g. the Internet) but at the same time no information can be extracted from the protected network. These devices are often used in the defense and infrastructure environments where security is critical. Many of them are compliant with rigorous Critical Infrastructure Protection (CIP) standards or regulations established by such organizations as NRC (U.S. Nuclear Regulatory

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_35,

Commission), NIST (National Institute of Standards and Technology) or NIAP (National Information Assurance Partnership).

The data diode, like other IT devices, can be additionally checked and evaluated by an independent body according to a stringent security standard – Common Criteria (CC) also known as ISO/IEC 15408 [6-8]. In the result of evaluation a certificate can be issued stating that the target of evaluation (TOE, i.e. a device) fulfilled all requirements congruent with the given Evaluation Assurance Level (EAL). This level tells a user how reliable the security measures built in the evaluated product are. On the EAL basis the user can decide whether to accept the risk of exposing the product to threats or not. According to the standard there are seven levels ranging from EAL1 to EAL7. Most of the certified IT products have middle-ranged levels (EAL3 – EAL4) due to the rising costs of evaluation and development processes at higher levels [9]. However, data diodes have very often higher EALs (EAL4+ or EAL7+) because of their relatively simple hardware structure that does not significantly increase the costs of the evaluation process [10], [11].

The developers have to prepare not only a product itself for the evaluation process but also some special accompanying evidence documents. These documents are key factors of successful evaluation results. That is why preparing them in a proper manner is a big challenge for developers not familiar with the rules of the Common Criteria. So far relatively little attention has been paid by researchers to support and make this task easier. Later in this paper it will be demonstrated how to help developers in preparing a part of a Security Target (ST) document by using a software tool made by the Institute EMAG in one of R&D projects [12].

The Security Target is the most important document which, in general, specifies "what is to be evaluated". ST is a security specification on a relatively high level of abstraction which describes how security measures should work and how reliable they are. These measures are presented by SFRs (Security Functional Requirements) and SARs (Security Assurance Requirements) components which are a kind of semiformal language of the Common Criteria standard. Thanks to the CC components all documents can be built in the same way by different developers for different types of products. The documents can be also verified according to one Common Evaluation Methodology (CEM) [13].

The CEM methodology defines evaluation activities and work units to be done by evaluators in order to issue verdicts about security measures implemented in a product and described in evidence documents. As a result, the evaluators can assess the documents in a relatively easy way because they have the ready-to-use evaluation methodology. But what about the developers? In contrast to the evaluators, they have not got too many options for help. Even the standard itself, very extensive, with lots of details and hundreds of components, is very difficult to use by inexperienced users. Of course there are many consulting companies on the market but their services are very expensive. Some possibilities of additional costs mitigation were proposed and described in the previous papers [14], [15]. The design patterns of evidence documents and computer tools were presented there as the way to support developers in elaborating necessary TOE documentation.

The literature review and state of the art showed that there were very few solutions which could help developers to fulfill all the requirements of the standard. Until now achieving the CC requirements has been still a very difficult and time-consuming task. There were also many guidelines, supporting documents with hints and tips and even templates of ST and other evidence documents [16-20]. Although such an approach improves developers' work, it results in an unacceptable number of analyses of hundreds of the CC components that have to be done by the developers on their own. That is why using only guidelines is still inconvenient as such. The situation has not changed too much for better. There were only a few software tools supporting elaboration of the CC documents as it was also mentioned in [21], [22]. That is why a few years ago the CCMODE (Common Criteria compliant, Modular, Open IT security Development Environment) R&D project was launched by the Institute of Innovative Technologies EMAG [23] to improve the developers' position.

The results of the project are methodology and software tools for developing and managing development environments of IT security-enhanced products for the purposes of their future evaluation and certification according to CC requirements. The software tool – called CCMODE Tools – integrates design patterns with documents generator application, knowledge base, evaluation methodology, and external supporting software dedicated for testing, flaws management and security analyses. The whole system can be used for a wide range of IT software and hardware devices or systems. The tool, along with the design patterns, was validated in a few projects concerning intelligent sensors [24], motion sensors [25], intelligent gas sensors for coal mines [26], and biometric systems [27]. The main features of the tool were presented in paper [28] whose main conclusion was that the design patterns supported by the software tool really facilitate and speed up development process and improve the quality of evidence documents.

Poland has not signed the Common Criteria Recognition Arrangement yet which allows to mutually recognize the CC certificates between the member states of this agreement. This is why the application of the CC standard is not obligatory in Poland. Yet there are some producers of special IT products (including data diodes) for military purposes who want their devices to be additionally accredited according to the CC requirements. One of the Polish producers of data diodes has started cooperation with the EMAG Institute in order to prepare a data diode to the CC accreditation process. As a result, an excellent opportunity emerged to make validation of the CCMODE Tools and to elaborate the trial version of the CC evidence documents of the product for the accreditation process at the same time. This example of cooperation shows that there could be a need for the future applications of the CC standard in Poland. The accreditation can be conducted by such Polish institutions as the Ministry of National Defense, the Internal Security Agency or Military Counterintelligence Service.

The purpose of this paper is to describe and examine a special plug-in application (another module of the CCMODE Tools system) dedicated to support security analyses of a product. This plug-in helps developers in elaborating the main parts of the ST document: a security problem definition (SPD) section which describes threats to product assets; a security objectives section which consists of statements resolving the

given security problem; a security requirements section which is presented in the semiformal language of SFRs and SARs components; a product summary specification containing security functions which should be implemented into the device in order to fulfill all security functional requirements. On the example of the data diode it will be demonstrated, in the results section of this paper, that the plug-in is an efficient and easy to use graphical tool for analyzing the security needs of the device and for elaborating the chosen sections of the ST document.

The paper is organized as follows. Section 1 presents the background of the Common Criteria standard and state of the art, and the main features of the data diode device. Section 2 describes three basic processes of the CC methodology and the general structure and features of the plug-in application. Section 3 delivers the results of the plug-in validation on the example of the data diode and presents main objects used to elaborate the chosen units of the ST document. Section 4 contains the results discussion and conclusions and states possible future work.

2 Methodology and Tools

The first possible way to create a new ST document is using a Protection Profile (PP) for the given type of products. If there is no suitable PP then a developer can use a few guidelines and templates issued by German Federal Office of Information Security (BSI) – the informal leader of researches in the field of the CC standard. This help documentation gives some advice on the structure and contents of evidence documents but developers have still many editorial activities left on their side.

The next two subsections of the paper describe respectively basic processes of the CC standard and the development stages of the plug-in software tool.

2.1 The Common Criteria Methodology for a Data Diode

Data diodes are products which can gain additional security assurance by applying the CC methodology. The positive evaluation results and certificate can be achieved by carrying out three basic processes: IT security development, TOE development, and IT security evaluation. These processes and the ST structure were described in detail in previous papers, e.g. [27], [28]. In the current paper, which concerns the range of the plug-in software support, only the most important activities of the security development process and the chosen sections of ST will be referred to and described. In the following paragraphs there were used some parts of the ST document for the CC certified data diode according to EAL4+ (called for short FFHDD – Fort Fox Hardware Data Diode) [10].

The IT security development process is based on security analyses whose purpose is to identify a product security problem and to resolve it by security objectives. That solution is described in the Security Target document in terms of security functional requirements (SFRs) which describe how security measures of the TOE should work to effectively counter the identified threats. In order to facilitate choosing the most suitable functional requirements for the given security objectives, the SFR

components are divided into 11 classes referring to such security issues as: security audit, communication, cryptographic support, user data protection, identification and authentication, security management, privacy, protection of the TSFs (TOE security functions), resource utilization, TOE access, trusted path/channels. These functional security requirements are next implemented into the device in the form of security functions during the TOE development process. These functions are also developed according to the security assurance requirements (SARs) of the claimed EAL.

As a result of the IT security development process the ST document is worked out. The complete document consists of the following sections: an ST introduction, a conformance claim, security problem definition (SPD), security objectives, extended component definition (optional), security requirements (SFRs and SARs), TOE summary specification. The plug-in software tool supports developers in elaborating all sections of the ST apart from the first ones: introduction and conformance claim which can be written in a standard text application. At the beginning of the IT security development process the security problem should be defined.

The SPD section defines the security problem that is to be addressed. It should be noted that usefulness of the ST document strongly depends on the quality of SPD and this is why it is worth to spend more time and resources to derive this section in a careful and reliable manner. What is more, the deriving process of SPD is outside the scope of the CC standard so developers can only follow the examples of the previously certified products to check how to make properly this section of the ST document. Here the plug-in software tool offers the developers very useful wizards and inferring mechanisms which make this process easier and quicker. The typical SPD consists of the following units: assets, threats, organizational security policies (OSPs), and assumptions.

The security problem is understood as the danger of assets loss. So it is important to precisely identify all assets that have to be protected by the TOE and to identify all entities which interact with the TOE. For the above mentioned example of the certified data diode – FFHDD the protected asset is the sensitive information on the High Security Level network. The entities which can use this information can be authorized or not, and can be users or processes e.g.: an administrator, a user, a hacker, an update service. The threat unit describes the threats that are to be countered by the TOE, its operational environment, or combination of the two. A threat definition consists of an adverse action performed by a threat agent on an asset. In the example of the FFHDD device the threat T.TRANSFER was defined as “A user or process on the High Security Level network that accidentally or deliberately breaches the confidentiality of some High Security Level information by transmitting data through the TOE to the Low Security Level network”. The next unit of ST shows OSPs which are security rules, procedures, or guidelines imposed by an organization to the TOE or to the operational environment of the TOE. No OSPs were defined for the FFHDD product but an example of OSP can be a statement “Only users with system administrator privilege shall be allowed to set up and manage the data diode connection system”. The last unit of ST shows assumptions that are made on the operational environment in order to make it secure enough for the TOE. In such an environment the TOE is able to provide all its security functionalities without disruptions. Assumptions can be

made on physical, personnel and connectivity aspects of the operational environment. In the case of FFHDD the following assumptions were defined: A.PHYSICAL "The intended operation environment shall store and operate the TOE in accordance with the requirements of the High Security Level side"; A.NETWORK "The TOE is the only method of interconnecting the Low Security Level network and High Security Level network. This prevents a threat agent from circumventing the security being provided by the TOE through an untrustworthy product". Once the complete SPD is defined the security objectives section of ST can next be elaborated.

The security objectives are the concise statements of the intended solution to the given problem defined by SPD. The security problem can be solved by two sets of objectives for the TOE and for the operational environment of the TOE. The first set shows what the TOE should do in order to protect the assets, e.g. for FFHDD: O.CONFIDENTIALITY "The information on the High Security Level side destination is kept confidential from the Low Security Level source". The second set shows the goals that the operational environment should achieve, e.g. for FFHDD: OE.PHYSICAL "The intended operation environment shall be capable of storing and operating the TOE in accordance with the requirements of the High Security Level side"; OE.NETWORK "The TOE is the only method of interconnecting the Low Security Level network and High Security Level network". In the end a developer should provide a rationale that security objectives counter all threats, enforce OSPs and upheld all assumptions. The next step is to translate the security objectives into the CC language of SFRs components.

The security requirements unit of ST consists of two groups of requirements: the security functional requirements (SFRs) – a translation of the security objectives for the TOE into a standardized, semiformal CC language; the security assurance requirements (SARs) – a description of how assurance is to be gained that the TOE meets the SFRs. The first group is made up from functional components taken from Part 2 of the CC standard and the second group are assurance components taken from Part 3 of the CC standard according to the claimed EAL (here EAL4+ for FFHDD). The SARs determine the range and details of the TOE development and security evaluation processes which are not in the focus of this paper. The SFRs are next implemented in the TOE security functions. In the example of FFHDD two functional components were selected which fulfill the TOE security objective for confidentiality: FDP_IFC.2 Complete Information Flow Control (FDP class – user data protection, IFC family – information flow control policy); FDP_IFF.1 Simple Security Attributes (IFF family – information flow control functions). Generally speaking these components permit the TOE security function to transmit sensitive information only from input (Low Security Level network) to output (High Security Level network) and not vice versa. The choice of proper SFRs is very difficult and depends on the developer's experience and knowledge of the CC standard. Therefore there could be a vast potential demand for supporting this part of ST elaboration. The security requirements elaboration is finalized by their rationale. Next these requirements should be implemented in the TOE security functions which are described in the last section of ST – TOE summary specification (TSS).

The TSS section provides potential consumers of the TOE with a general technical description of how the product satisfies all the SFRs. In many cases the SFRs are gathered within one security issue, e.g. for FFHDD it is the user data protection class (FDP) described in the Part 2 of the CC standard. In the ST of this data diode is indicated that confidentiality of information is provided by using two functional requirements responsible for information flow control.

The IT security development process provides a set of security functions, which should be implemented into the TOE at the assurance of claimed EAL. The process of deriving security functions can be facilitated by the use of the plug-in software tool which was elaborated in the CCMODE R&D project.

2.2 Building the Plug-in Module

One of the CCMODE project results is the CCMODE Tools system. The system integrates: modules of the development environment management, design patterns, knowledge base, evaluation methodology, and external supporting software. Among the external applications there is the Enterprise Architect (EA) of the Sparx platform which was used as a basis for modeling, development and security analyses made in UML (Unified Modeling Language). In the CCMODE project a special plug-in for EA was worked out. The plug-in uses basic features of EA system and UML in order to support users in defining the TOE security problem and next in solving that problem by selecting security objectives and functional security requirements (SFRs).

At the beginning of the CCMODE project the survey of UML software development environments was made. As a result, EA was chosen as the most convenient and easy platform for third-party software implementations. The EA system is a rather cheap solution, known by many IT developers, which enables implementation of additional plug-ins in the form of DLL (Dynamic-Link Library) files. The EA system uses a database which is open and accessible to programmers so it is quite easy for them to build their own tables for the purposes of potential plug-ins. The EA producer also expressed preliminary agreement to offer the plug-in for their customers.

Firstly, the model of the plug-in was made in UML on the basis of functional requirements proposed by IT products developers. The database structure was modeled for the ST documents sections data. The toolbox of graphical objects for building elements of SPD and ST sections was worked out. The toolbox contains objects of assets, security objectives, assumptions, threats, security functions and requirements, and connectors for making relations between them. The objects have dialog windows which are connected to the knowledge base of the CCMODE Tools system. On that basis special wizards windows and inferring procedures were worked out.

Next these procedures use predefined ST elements stored in the knowledge base, e.g. assets types and forms with corresponding threats, threats and corresponding objectives, and objectives connected to predefined SFRs. The procedures use many other parameters of the objects, for instance: the form and type of assets which influence the type of adverse actions and possible vulnerabilities, value of assets, possible loss or the likelihood of a dangerous security incident. The knowledge base comprises the guidelines that help to resolve typical security problems with the use of predefined

security objectives, threats, assumptions, and security policies. The predefined elements of SPD and its solutions are connected by relations through the database in order to make the lists of proposed security objectives and SFRs shorter. These relations are the result of analyses made in the CCMODE project concerning dozens of STs and PPs, templates and guidelines describing how to define and solve the security problem. The last steps of the plug-in development process were verification and validation of a prototype software tool in some ST projects of IT devices.

The next section of the paper presents some final results of the plug-in application for elaborating the ST document of the chosen data diode (FFHDD).

3 Results of Plug-in Application

The plug-in tool was used for making the SPD section of ST for the chosen data diode. As a result, the graphical form of SPD and its solution (described in section 2.1 of this paper) were worked out (Fig. 1).

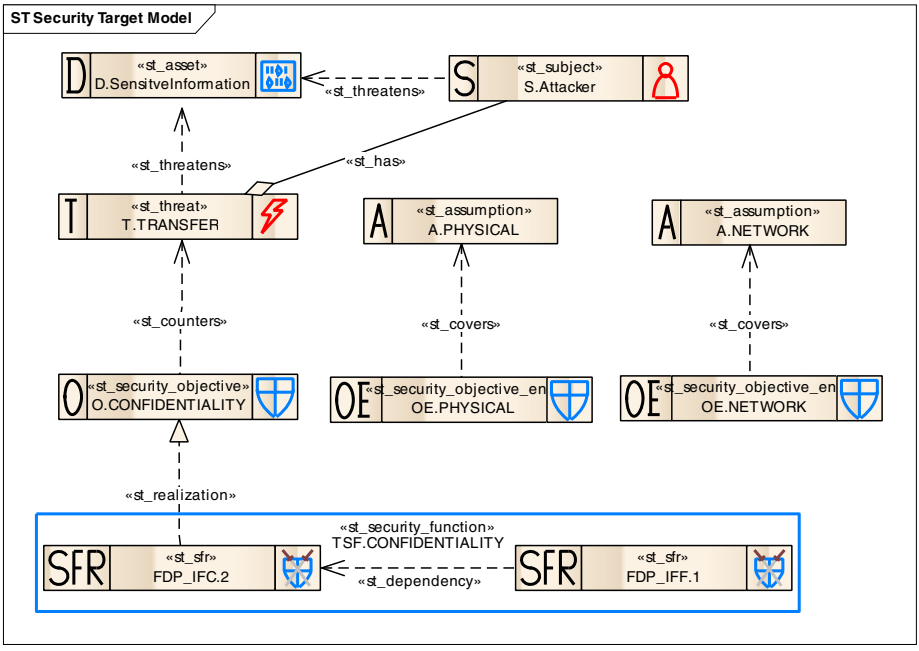


Fig. 1. The ST model of the data diode

Figure 1 shows all major objects of the plug-in ST toolbox with their mutual relations: D – assets, S – entities using assets, T – threats, A – assumptions, O – security objectives for the TOE, OE – security objectives for the operational environment of the TOE, SFR – security functional requirement, and the blue line which groups the selected SFRs into one security function. This graphical form of ST is next automatically

pasted to the final ST document in another module of the CCMODE Tools system. This module is called documents generator and it uses the design pattern of ST.

4 Conclusions

This paper presented the software tool to support developers in elaborating the security problem definition – the most important part of the Security Target document. The plug-in tool was one of the results of the CCMODE R&D project. The tool is a part of the bigger CCMODE Tools system which fully supports security development and TOE development processes.

The paper presents general guidelines for the data diodes developers in accordance with the Common Criteria standard. It was noticed that some Polish developers might need to accredit similar devices according to the CC standard, that is why the data diode was used as an example. The paper provides developers with basic information about the security development process which can be supported by the plug-in application.

Therefore the plug-in software tool as well as the whole CCMODE Tools system are promising means for supporting developers of IT security enhanced products. The plug-in facilitates and speeds up the IT security development process and improves the quality of the ST document in the sections concerning security problem to be solved by the TOE. The objects used in the plug-in for one device can be reused in other projects of similar products. The developer gets a graphical picture of the security problem which is easy to analyze and to verify. The selection of security objectives and SFRs is supported by inferring wizards based on the knowledge base. Each step of SPD development can be checked and verified by the reports module of the plug-in. The reports consist of set of tables which have a similar role to rationale tables used in standard evidence documents but this the topic for the next paper.

Future work will be focused on building a standalone, independent plug-in application with its own database in order to separate it from the CCMODE Tools system. As a result the mobility of the plug-in can be improved and the costs of deployment and implementation can be reduced.

References

1. Fort Fox Hardware Data Diode, https://www.fox-it.com/en/products/datadiode
2. Filbico Data Diode, http://www.filbico.pl/index.php/pl/zno
3. Waterfall Security Solutions USA, http://www.waterfallsecurity.com/technology
4. Owl Computing Technologies, http://www.owlcti.com/dualdiode_technology.html
5. VS-diode, http://www.genua.eu/produkte/datendiode/index.en.html
6. Common Criteria for Information Technology Security Evaluation (Ver. 3.1, Revision 4) Part 1: Introduction and general model (ISO/IEC 15408-1) (September 2012)
7. Common Criteria for Information Technology Security Evaluation (Ver. 3.1, Revision 4) Part 2: Security functional requirements (ISO/IEC 15408-2) (September 2012)

8. Common Criteria for Information Technology Security Evaluation (Ver. 3.1, Revision 4) Part 3: Security assurance requirements (ISO/IEC 15408-3) (September 2012)
9. The Common Criteria Portal, http://www.commoncriteriaportal.org
10. Certification Report EAL4+ – NSCIB-CC-09-11025-CR, Fort Fox Hardware Data Diode, version FFHDD2+, TNO certification (September 3, 2009)
11. Certification Report EAL7+ – NSCIB-CC-09-11025-CR2, Fort Fox Hardware Data Diode, version FFHDD2+, TNO certification (June 16, 2010)
12. Common Methodology for Information Technology Security Evaluation (Version 3.1, Revision 4). Evaluation Methodology (September 2012)
13. Białas, A.: Patterns Improving the Common Criteria Compliant IT Security Development Process. In: Zamojski, W., Kacprzyk, J., Mazurkiewicz, J., Sugier, J., Walkowiak, T. (eds.) Dependable Computer Systems. AISC, vol. 97, pp. 1–16. Springer, Heidelberg (2011)
14. Rogowski, D., Nowak, P.: Pattern Based Support for Site Certification. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J., et al. (eds.) Complex Systems and Dependability. AISC, vol. 170, pp. 179–193. Springer, Heidelberg (2012)
15. Białas, A. (ed.): Applying design patterns in security development process according to the Common Criteria standard. Original Polish title: Zastosowanie wzorców projektowych w konstruowaniu zabezpieczeń informatycznych zgodnych ze standardem Common Criteria. Wydawnictwo Instytutu Technik Innowacyjnych EMAG, financed by UE POIG 1.3.1, Katowice (2011) ISBN 978-83-932737-2-0
16. ISO/IEC TR 15446: Information technology – security techniques – guide for the production of Protection Profiles and Security Targets (2009)
17. BSI: The PP/ST guide, Version 1, Revision 6.2 (August 2007)
18. BSI: Guidelines for developer documentation according to CC Version 3.1 (2007)
19. BSI: Guidelines for evaluation reports according to Common Criteria Version 3.1, Version 2.00 for CCv3.1 rev. 3 (2010)
20. Higaki, W.H.: Successful Common Criteria evaluations. A practical guide for vendors. Create Space Independent Publishing Platform (2010)
21. Rogowski, D.: Computer-aided tool based on Common Criteria related design patterns. In: Korczak, J., Dudycz, H., Dyczkowski, M. (eds.) Business Informatics, vol. 3(29), pp. 111–127. Wrocław University of Economics Research Papers (2013)
22. CCMODE Project (Common Criteria compliant, Modular, Open IT security Development Environment), http://www.commoncriteria.pl/
23. Białas A.: Security-related design patterns for intelligent sensors requiring measurable assurance. Electrical Review (Przegląd Elektrotechniczny) 85(R.85(7)), 92–99 (2009) ISSN 0033-2097
24. Bialas, A.: Common Criteria Related Security Design Patterns for Intelligent Sensors – Knowledge Engineering-Based Implementation. Sensors (August 2011)
25. Białas, A.: Ontological approach to the motion sensor security development. Electrical Review (Przegląd Elektrotechniczny) 85(R.85(11)), 36–44 (2009) ISSN 0033-2097
26. Bialas, A.: Common Criteria Related Security Design Patterns – Validation on the Intelligent Sensor Example Designed for Mine Environment. Sensors (April 2010)
27. Bialas, A.: How to develop a biometric system with claimed assurance. In: Proceedings of the 2013 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 775–780. IEEE Xplore Digital Library (2013)
28. Rogowski, D.: Software Implementation of Common Criteria Related Design Patterns. In: Proceedings of the 2013 Federated Conference on Computer Science and Information Systems (FedCSIS), Annals of Computer Science and Information Systems, pp. 1147–1152. IEEE Xplore Digital Library (2013)

Increasing Performance of SMS Based Information Systems

Mariusz Rychlicki and Zbigniew Kasprzyk

Warsaw University Of Technology, Faculty of Transport,
75 Koszykowa St, 00-662 Warsaw, Poland
{mry,zka}@wt.pw.edu.pl

Abstract. How do information systems influence and permeate the world and the society is the light motif of this paper. The SMS service is so simple yet so powerful what was illustrated in this paper as were example implementations. The importance of system efficiency was investigated. Laboratory experiments were performed to analyse how efficient are those systems. Relationships were derived and activity ratio determined that show how system performance could be improved through analysis and synthesis of an example information system using SMS service.

Keywords: information system, performance improvement, sms, transport.

1 Introduction

Modern world hinges on efficient circulation and processing of information as do modern information societies. Information has been commodified. It has been given value, is subject to exchange and is even being sold. Modern businesses from technology as well as other sectors rely on it to operate. A strong case in point is modern transport, where information about e.g. current location of means of transport is key to successfully run transport processes. In order to address demand for efficient and fast information exchange, two telecommunications services are provided: electronic mail enabling users to communicate using text and image as well as SMS (Short Message Service) enabling sending text messages over digital GSM cellular networks.

SMS services have been delivered for over 20 years now almost in identical format. It is a telecommunications phenomenon. Ian Pearson was the first to conceptualise communication based on short text messages back in early 90's of the XX century. His vision materialised already on December 3rd 1992, when in Great Britain a Vodafone employer by the name of Neil Papworth sent Christmas greetings to work colleagues via SMS. Enthusiastic reception of new communication form encouraged the company to introduce a new service [1]. Originally, it was intended to serve as a notification tool, providing status updates about the network, downtimes etc. No one expected, that fairly straightforward service would revolutionise the telecommunications industry. The extraordinary success of SMS messaging is blatant. It was down to its simplicity, speed and reliability. Therefore it is hardly surprising it manages to rein-

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_36,

vent itself time and time again in new areas of technology, especially thanks to introduction of the SMPP protocol (Short Message Per to Peer Protocol) [2]. Information systems are one of those areas, where many sectors (e.g. transport) benefit vastly from the speed and simplicity of obtaining information in that manner.

2 SMS in Information Systems

In order to become entirely (to the maximum possible extent) independent from external providers, new solutions have to be devised enabling easy SMS implementation and integration with information systems. Those solutions necessitate dedicated, highly customisable system software. The ActiveXperts SMS Messaging Server framework class software provides just that [3]. It is dedicated for sending, receiving and processing SMS and email messages as well as creating own applications using those functions and mechanisms. Combined with hardware (server, GSM modem) and database access that software enables creating information systems performing various informational functions e.g. query-response. Functional diagram of such system was illustrated in figure 1.

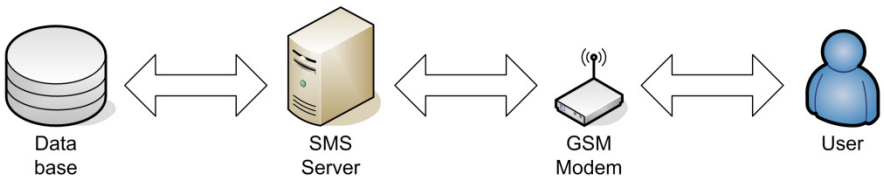


Fig. 1. Basic functional diagram of information system

A real life example would be e.g. a system informing about border wait times at crossing points situated along Polish Eastern border. For logistics contractors as well as lorry drivers it is a very important parameter which directly influences transport and logistics processes. In order to meet expectations of all of those stakeholders, Customs Chamber in Bialystok publishes publicly that data via Web Service. It enables to automatically download data from the server, without any input from the user. That data may be used to compute optimum route, analyse traffic passing through border checkpoints etc. Furthermore, the scope of data was expanded by providing information about traffic vehicle type: passenger cars, lorries and coaches [4]. The scope and easy access to the data means that implementation of such information system comes easily using the aforementioned hardware and system software ActiveXperts SMS Messaging Server. Neither an Internet access is required nor visiting any particular website. The system requires inputs via SMS in predetermined format and returns wait times in reply message. Fig. 2 shows an example query and its processing through the system. SMS message reading PL.UA queries about border wait times at Polish border to cross into Ukraine. The reverse UA.PL means the opposite crossing direction.

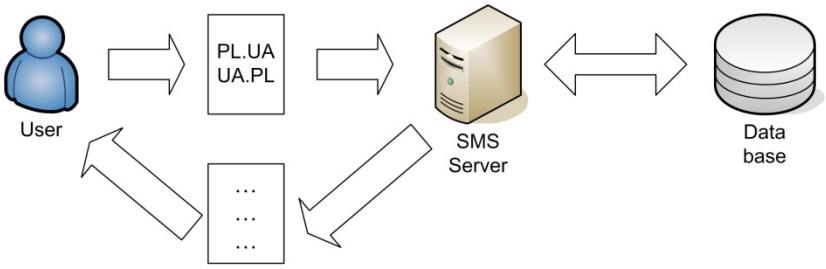


Fig. 2. Example SMS query submitted to the information system.

There are far more examples including e.g. a system informing about petrol prices in given region of the country. They all could be described as simple, flexible and easy to access - key features in transport. Hence those solutions are so interesting and worthwhile. Their capability to process given number of queries (SMS messages) in a time period defined as system performance is a key metric as well as system utilisation. Of course, efficiency related issues are not the sole concern when designing information systems. Among the most important factors, especially in rail transport are influence of electromagnetic interference (mainly low frequencies) [5] and system reliability in relation to its structure [6].

3 SMS Query Processing Times

Authors' previous body of work [7] presented detailed measurement results obtained when testing an SMS based information system. The purpose of those measurements was to determine turnaround for processing SMS queries using standard-issue GSM modems. Another condition, aimed at cost cutting and increasing flexibility, was using solely a Polish cellular service provider without any intermediary providers and not using the SMPP protocol. The test involved sending 10, 20, 50 and 100 SMS messages via different mobile operators (Orange, Plus, Play, Heyah), different GSM modems and at different times of day. Example and partial measurement results, here meaning SMS sending times, were collated in table 1.

Test results did not prove turnaround times varied depending on amount of SMS messages per batch, neither they did across GSM networks. Sending time was noticeably slower, depending on SMS batch size, only in case of Orange mobile operator. Message chronology was disrupted only on the Orange network as well (each SMS message was tagged with a unique ID). This issue was not observed for other cellular networks. Test results served to determine boundary values and expected turnaround times, which should be assured by the analysed SMS messaging system.

Table 1. Mean sending times of SMS queries via GSM network

GSM network	Orange	Plus	Play	Heyah
Signal strength	51%	55%	52%	53%
SMS batch	t_{mean}	t_{mean}	t_{mean}	t_{mean}
10	2.200	2.300	2.000	2.400
20	2.175	2.200	2.125	2.100
50	2.730	2.290	2.030	2.200
100	2.770	2.150	2.065	2.185

4 Information System Performance

Note that the information system designed using the aforementioned guidelines resembles in terms of operating principle a central server. It was well documented in papers [8, 9] and substantiated by many examples. The information system in question consists of a SMS server, operating as a central unit S_1 controlled by processor P_1. The server is connected with GSM modems i.e. input/output units S_i which are also controlled by built-in but less powerful processors P_i (i = 2, 3, ..., M). Modems are receiving requests (SMS messages) which are fed to the server for processing. The server receives SMS message, queries database, formats the response and feeds to a GSM modem. The server and processor P_1 additionally run their operating systems and database, therefore they need to be equipped with sufficient computing power and have a necessary operating window. This approach allows to view the analysed information system as a generalised model of closed exponential network. Figure 3 shows its diagram.

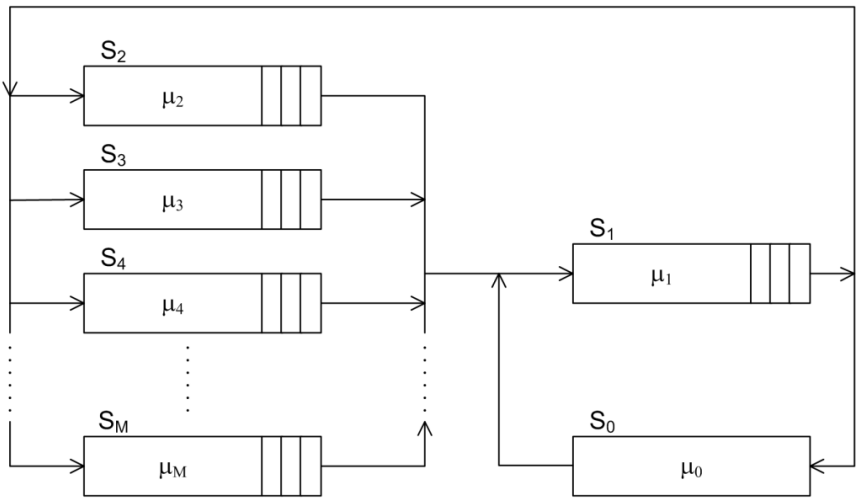


Fig. 3. Diagram of the generalised information system model

It was assumed that a SMS message could not be lost due to busy GSM modem - GSM network would not release a message when busy and its memory buffer is, as far as information system is concerned, unlimited. SMS processing times obtained by processor P_i of each modem are statistically independent of each other and are described by exponential distribution μ_i. The SMS queue is organised by the FIFO method. The server processes every SMS message for a given time after which it generates a response for the modem with probability p_i and moves to processing another SMS message. Mean processing times are denoted as follows

$$\frac{1}{\mu_i} \text{ , where i = 1, ..., M} \tag{1}$$

Activity ratio of i-th modem Ui ($0 \leq Ui \leq 1$) which determines in which part of time interval modem is busy processing incoming SMS message. As per Chang-Lavenberg theorem [9]

$$U_i = \rho_i \text{ , where i = 1, ..., M} \tag{1}$$

and

$$\frac{U_i}{U_j} = \frac{\rho_i}{\rho_j} = \frac{\beta_i}{\beta_j}\frac{\mu_i}{\mu_j} = \frac{p_i}{p_j}\frac{\mu_i}{\mu_j} \text{ , where i = 1, ..., M} \tag{3}$$

Thus U_i/U_j is independent of streams x_i and x_j, therefore for a modem with one processing channel for any x

$$U_i = \frac{\beta_i x}{\mu_i}\frac{G(N-1,\ x)}{G(N,\ x)} \tag{4}$$

Thus

$$\gamma_i(x) = \frac{\beta_i x}{\mu_i} \text{ , where i = 1, ..., M} \tag{5}$$

and $x = x_1$

$$\gamma_i(x_1) = \rho_i \text{ , where i = 1, ..., M} \tag{6}$$

Mean number of requests stacked in modem in steady state.

$$\overline{n_i} = \sum_{k=1}^{N} k\, p(n_i = k) = \sum_{k=1}^{N} p(n_i \geq k) \tag{7}$$

where

$$\mathrm{p}(n_i \geq k) = [\gamma_i(x)]^k \frac{G(N-k,\ x)}{G(N,\ x)} \tag{8}$$

and

$$\mathrm{G}(N, x) = \sum_{S(N,M)}\{\prod_{i=1}^{M}[\gamma_i(x)]^{n_i}\} \tag{9}$$

$$\overline{n_i} = U_i \tag{10}$$

where U_i is the i-th modem's activity ratio

Because

$$U_i = 1 - p(n_i = 0) = p(n_i \geq 1) \tag{11}$$

then for k = 1 the relationship (8) yields

$$U_i = \gamma_i(x)\frac{G(N-1,\ x)}{G(N,\ x)} \text{ , where i = 1, ..., M} \tag{12}$$

for $U_i = \rho_i$ and

$$G(N-1,x) = G(N,\ x) \tag{13}$$

the following relationship is derived

$$g(n,m) = g(n,\ m-1) + \gamma_m(x)g(n-1,\ m) \text{ , where m > 1 and n > 0} \tag{14}$$

and

$$\sum_{i=2}^{M} p_i = 1 - p_1 \tag{15}$$

Table 2. Quantities g(n,m), [$n \geq 0$, $U_i(n_i) < U_{gr}$] were computed given i = 1, 2, 3, 4

	γ_1 =1.000	γ_2 =1.000	γ_3 =1.000	γ_4 =1.400	
n/m	1	2	3	4	G(n,20)
0	1.00	1	1	1	G(0,20)
1	1	2	3	4.40	G(1,20)
2	1	3	6	12.16	G(2,20)
3	1	4	10	27.02	G(3,20)
4	1	5	15	52.83	G(4,20)
5	1	6	21	94.97	G(5,20)
6	1	7	28	160.95	G(6,20)
7	1	8	36	261.34	G(7,20)
8	1	9	45	410.87	G(8,20)
9	1	10	55	630.22	G(9,20)
10	1	11	66	948.30	G(10,20)

The following quantities were assumed for considering information system based on SMS server, three GSM servers (hence M = 4) and ActiveXperts SMS Messaging Server on the grounds of previous results [7]: $\mu_1 = 20$, $\mu_2 = 2$, $\mu_3 = 2$, $\mu_4 = 5$, $p_1 = 0.45$, $p_2 = 0.10$, $p_3 = 0.10$, $p_4 = 0.35$. Using above values, maximum system processing capability (number of SMS messages N_1) was determined given the activity ratio U_i of each processor P_i was lower than boundary value U_{gr}. Obtained results were tested with the following condition (16)

$$\sum_{i=1}^{4} \overline{n_i} = N_1 \tag{16}$$

where $\overline{n_i}$ determines mean number of messages processed per unit of time.

From (15) we get

$$\sum_{i=2}^{4} p_i = 1 - p_1 = 0.55 \tag{17}$$

Given x = μ_1 = 20 we get from relationship (5) $\gamma_1(20) = 1.00$, $\gamma_2(20) = 1.00$, $\gamma_3(20) = 1.00$, $\gamma_4(20) = 1.40$. The quantities g(n,m) and activity ratio $U_i(n)$ were computed using (12) and (15). Table 2 and 3 present obtained results.

Table 3. The activity ratio was computed $U_i(n)$, $[n \geq 0, U_i(n_i) < U_{gr}]$ given i = 1, 2, 3, 4

	γ_1 =1.000	γ_2 =1.000	γ_3 =1.000	γ_4 =1.400
n/ U_i	U_1	U_2	U_3	U_4
0	---	---	---	---
1	0.227	0.227	0.227	318
2	0.362	0.362	0.362	0.507
3	0.450	0.450	0.450	0.630
4	0.511	0.511	0.511	0.716
5	0.556	0.556	0.556	0.779
6	0.590	0.590	0.590	0.826
7	0.616	0.616	0.616	0.862
8	0.636	0.636	0.636	0.890
9	0.652	0.652	0.652	0.913
10	0.665	0.665	0.665	0.930

Analysis of obtained results aims to find maximum N_1, satisfying $\forall_i \in \{1, 2, 3, 4\}$, $U_i(N_1) < U_{gr}$. Given $U_{gr} = 0.80$ for obtained results the condition is satisfied for $N_1 = 5$. Then, using (12) and (10) $U_i(n)$ were determined and $\overline{n_1} = 1.026$, $\overline{n_2} = 1.026$, $\overline{n_3} = 1.026$, $\overline{n_4} = 1.923$ were computed. Note that (16) is satisfied proving obtained results correct. It was assumed that to guarantee system stability all processors P_i were under uniform load. Table 3 shows that ratio U_4 is significantly different from remaining activity ratios, U_1, U_2 and U_3. Probabilities p_i are sought for to satisfy

$$U_1 = U_2 = U_3 \tag{18}$$

thus

$$\rho_1 = \rho_2 = \rho_3 \tag{19}$$

and

$$1 - p_i = p_2 + p_3 + p_4 \tag{20}$$

and then

$$p_1 = 1 - \sum_{i=2}^{4} p_i = 0.450 \tag{21}$$

$$p_2 = p_3 \frac{\mu_2}{\mu_3} = 0.122 \tag{22}$$

$$p_2 = \frac{\sum_{i=2}^{4} p_i}{\frac{\mu_2}{\mu_3}+1+\frac{\mu_4}{\mu_3}} = 0.122 \tag{23}$$

$$p_4 = p_3 \frac{\mu_4}{\mu_3} = 0.306 \tag{24}$$

Again, given $x = \mu_1 = 20$ for $p_1 = 0.450$, $p_2 = 0.122$, $p_3 = 0.122$ and $p_4 = 0.306$ we get from (5) $\gamma_1(20) = 1.00$, $\gamma_2(20) = 1.22$, $\gamma_3(20) = 1.22$, $\gamma_4(20) = 1.22$. The quantities g(n,m) and activity ratio $U_i(n)$ were computed again using (12) and (15). Table 4 and 5 present obtained results.

Table 4. Computing g(n,m), $[n \geq 0, U_i(n_i) < U_{gr}, U_2 = U_3 = U_4]$ for i = 1, 2, 3, 4

	γ_1 =1.000	γ_2 =1.222	γ_3 =1.222	γ_4 =1.222	
n/m	1	2	3	4	G(n,20)
0	1	1	1	1	G(0,20)
1	1	2.22	3.44	4.67	G(1,20)
2	1	3.72	7.93	13.63	G(2,20)
3	1	5.54	15.23	31.89	G(3,20)
4	1	7.77	26.39	65.36	G(4,20)
5	1	10.50	42.75	122.64	G(5,20)
6	1	13.83	66.09	215.97	G(6,20)
7	1	17.91	98.68	362.65	G(7,20)
8	1	22.89	143.50	586.73	G(8,20)
9	1	28.97	204.36	921.47	G(9,20)
10	1	36.41	286.18	1412.43	G(10,20)
11	1	45.50	395.28	2121.59	G(11,20)

Table 5. Determining activity ratio $U_i(n)$, $[n \geq 0, U_i(n_i) < U_{gr}]$ for i = 1, 2, 3, 4

	γ_1 =1.000	γ_2 =1.222	γ_3 =1.222	γ_4 =1.222
n/ U_i	U_1	U_2	U_3	U_4
0	---	---	---	---
1	0.214	0.262	0.262	0.262
2	0.342	0.418	0.418	0.418
3	0.427	0.522	0.522	0.522
4	0.488	0.596	0.596	0.596
5	0.533	0.651	0.651	0.651
6	0.568	0.694	0.694	0.694
7	0.596	0.728	0.728	0.728
8	0.618	0.755	0.755	0.755
9	0.637	0.778	0.778	0.778
10	0.652	0.797	0.797	0.797
11	0.666	0.814	0.814	0.814

Note that now the maximum system processing capability (number of SMS messages N_2) determined given the activity ratio U_i of each processor P_i lower than boundary value $U_{gr} = 0.8$ was $N_2 = 10$. Mean number of messages was $\overline{n_1} = 0.950$, $\overline{n_2} = 1.350$, $\overline{n_3} = 1.350$, $\overline{n_4} = 1.350$ respectively. Note that (16) is satisfied in this case as well thus proving again obtained results correct. Uniform load applied to each GSM modem allows to double the number of processed SMS messages thus increases system performance twofold.

5 Conclusions

In the world where exchange and processing of information is critical, data communications information systems play a vital role. Their flexibility and versatility combined with easy implementation on system level creates practically unlimited opportunities for application. Further conducive to that is fairly straightforward hardware and availability of system software with broad customisation capabilities. Combined with classic SMS services, it becomes a complete solution facilitating data exchange in query-response format. A number of issues, however, would need to be resolved prior to practical implementations. Some of them are performance related i.e. the capability to process a predetermined batch of information and queries submitted over a given time period. The activity ratio may serve as a parameter describing that performance. In the analysed case, it was proven that the ratio depends not only on the number of queries (messages) to process, but also their even distribution and allocation across input/output devices. If successfully carried out, processing capabilities are doubled thus increasing twofold performance of the entire information system. The difference in performance is so significant it shall be subject to further research aimed at determining possible practical implementations into real life information systems.

References

1. Hillebrand, F.: Short Message Service (SMS): The Creation of Personal Global Text Messaging. John Wiley & Sons, West Sussex (2010)
2. Short Message Peer to Peer Protocol Specification v3.4, SMPP Developers Forum (1999)
3. Product catalogue (December 16, 2013), http://www.activexperts.com/sms-messaging-server/
4. Opis usługi – udostępnienie danych o czasie oczekiwania na granicy RP (December 16, 2013), http://www.granica.gov.pl/nowa_usluga.php?v=pl
5. Paś, J., Duer, S.: Determination of the impact indicators of electromagnetic interferences on computer information systems. Neural Computing & Applications (2012), doi:10.1007/s, 00521-012-1165-1
6. Rosinski, A.: Design of the electronic protection systems with utilization of the method of analysis of reliability structures. In: Proceedings of the Nineteenth International Conference on Systems Engineering, ICSEng 2008, Las Vegas, USA (2008) 978-0-7695-3331-5

7. Rychlicki, M., Kasprzyk, Z.: Integracja usług poczty elektronicznej oraz sms w małych i średnich przedsiębiorstwach transportowych. Zeszyty Naukowe Politechniki Warszawskiej, Seria Transport, z. nr 92, 1230-9265. Oficyna Wydawnicza Politechniki Warszawskiej (2013)
8. Czachórski, T.: Modele kolejkowe systemów komputerowych. Politechnika Śląska, Gliwice (1999)
9. Filipowicz, B.: Modele stochastyczne w badaniach operacyjnych - analiza i synteza systemów obsługi i sieci kolejkowych. WNT, Warszawa (1996)

Internet-Based Production Monitoring and Reporting

Krzysztof Sacha and Wojciech Pikulski

Warsaw University of Technology, Warszawa, Poland
k.sacha@ia.pw.edu.pl, w.pikulski@elka.pw.edu.pl

Abstract. This paper describes a new approach to production data monitoring and reporting. The monitoring and reporting system called SMARP is composed of a small transponder, located on the plant floor, and a server, which can be located anywhere in the Internet. The main goal of SMARP is to provide the manufacturing decision maker with aggregated on-line process data in order to describe the effects of the plant operation, the effectiveness of the plant equipment and the causes of loses, such as accidents, damages and stoppages. The user of SMARP can also be the plant owner or any other authorized person, who can connect to the server through an arbitrary communication device, e.g. a laptop, a tablet or a mobile phone.

Keywords: production monitoring, SCADA, Internet based SCADA, manufacturing execution system.

1 Introduction

A typical plant consists of machines and devices arranged into production lines, and operating under the control of local controllers, which acquire data from the sensors and convey commands to the actuators of those machines and devices. Local controllers communicate through a computer network (a field bus) with supervisory controllers, which monitor and coordinate the plant operation. The main tasks of a Supervisory Control and Data Acquisition (SCADA) system are: Acquiring process data from the plant sensors, detecting alarms and abnormal situations, presenting the data to human operators and executing the operator commands. The scope of data and the way of presentation match the needs of an operator who controls the plant operation.

Control systems can be interfaced to an Enterprise Resource Planning (ERP) system, which looks at production orders and aggregates the process data to describe economic effects of the plant operation. SCADA server can be used as a gateway between the company's control and enterprise networks. The scope of data acquired and reported by ERP system match the needs of the manufacturing decision maker.

Typically, control systems reside on the plant floor and support on-line activities of the process operators, while the enterprise management systems reside on the main servers and support activities of the economic department of the company. The drawbacks are: High cost of the control and management systems, a dependence on particular vendors and restricted access to the process data, which is locked in a control room and accessible only to engineers trained to operate the proprietary systems.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_37,

The rest of the paper is organized as follows. Section 2 introduces SMARP system. Related work is discussed in Section 3. The architecture of SMARP server is shown in Section 4. An XML-based method for specifying the server is described in Section 5. A model of the server operation is introduced in Section 6. A discussion of the project status and the plans for future work are given in Conclusions.

2 Overview of SMARP

The described traditional SCADA-ERP architecture does not fit well the needs of the owners of small enterprises, who cannot afford such expensive systems, and who need a cheap solution with a possibility to monitor and control their businesses from a remote area. An alternative solution, much cheaper and more flexible than the traditional one, can be an Internet-based system for production monitoring, analysis and reporting (SMARP), which utilizes the public Internet infrastructure and offers a possibility to monitor and control the business aspects of the plant operation from anywhere in the world. The architecture of SMARP is shown in Fig. 1.

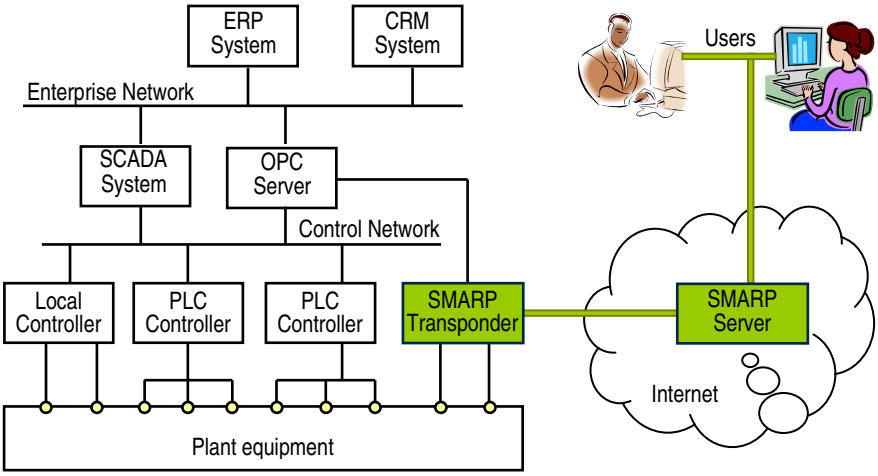


Fig. 1. The architecture of SMARP

The system consists of two basic components:

- SMARP transponder, which acquires the selected process data items;
- SMARP server, which stores, processes and presents the data to the users.

The transponder is a local device, which does not interfere with the plant operation. It acquires the process data from the controllers and additional sensors, and sends the data to the server through the Internet. The server is a computer which can be located anywhere in the network, e.g. on-site or in a cloud. The user of the server is the manufacturing decision maker, e.g., the plant owner, who can connect to the server through an arbitrary communication device, e.g. a laptop, a tablet or a mobile phone.

The advantages of SMARP architecture are the following:

- Low cost and ease of the system deployment.
- No need for individual programming of the server.
- High flexibility and the ability work in a cloud environment.
- Openness of the architecture and the communication standards.

3 Related Work

SCADA system provides the means to monitor and control plant devices from a central location. A typical system is deployed with proprietary software and hardware and with dedicated communication infrastructure. The technology is old and a general description can be found in many places, e.g. [1,2].

OPC is a technology developed by OPC Foundation in order to interface industrial controllers with popular PC computers. The first version (1998) was tied to Microsoft OLE and DCOM technologies, however, the latest version OPC UA (2012) is technology independent. A suit of standard specifications can be found in [3].

Internet SCADA (iSCADA) is a newer approach, characterized by the use of the Internet infrastructure to combine traditional SCADA design with the open communication protocols, services and data formats in order to deliver cost-effective SCADA solutions. The approach is still subject to research effort. One example is the system architecture described in [4,5], which consists of SCADA server and web server located at intranet, a firewall that separates intranet from the Internet, and a set of web clients. SCADA server supervises PLC controllers working over a control network and communicates with web server over the intranet. The clients can be located both: In intranet or in the Internet. They are computers, mobile phones or tablets, which use a standard web browser to access the web server. iSCADA solutions are offered by several vendors [6,7].

Our work fits partially into iSCADA approach; however, our goals are different. We do not want to build another SCADA system, to support process operators and to replace the existing systems. Instead, we intend to develop a production monitoring and reporting tool, to support the manufacture decision maker in controlling the economic effectiveness of the plant. Business scenarios can be the following.

The owner of a small enterprise that maintains a production line and a few auxiliary machines controlled by local controllers, neither needs, nor can afford the expensive SCADA system. Instead, he or she can deploy a simple transponder with few sensors, like photocells to register the products, and send the process data to a server located in the Internet cloud. The plant owner can rent an access service on subscription basis and pay small monthly installments, instead of paying for the entire investment. The service allows him to access the data by means of a web browser from an arbitrary place.

A huge enterprise with several production lines and full-fledged control and managements systems can maintain a proprietary SMARP server, installed on the plant floor. The system can serve management staff to analyze the overall equipment effectiveness (OEE), track and trace the products, and monitor the alarm conditions.

4 Architecture of SMARP Server

The conceptual architecture of SMARP server is shown in Fig. 2. The server receives a continuous stream of messages that convey process data sent by transponder (or transponders) attached to the plant installation. Messages can also be received from a touch panel, which can be attached to the transponder. An independent source of the server activity is a timer module, which counts time signals and triggers periodical computations within the server. All the input messages are temporarily stored in an input queue, from which they are fetched and processed in a sequential way.

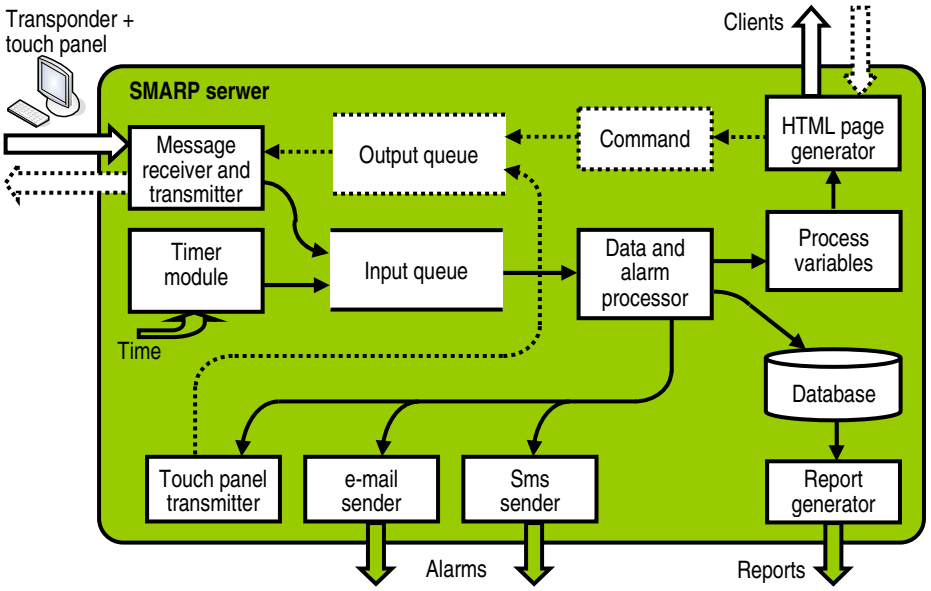


Fig. 2. The architecture of SMARP server

A single server can receive process data from several transponders, attached to the same or to different plants. Data related to the same plant are stored and processed jointly. Data related to different plants are treated separately. Thus, we can imagine that each plant has a separate instance of the server in our conceptual model.

The processing of data within the server consists of the following activities:

- Storage of the current values of measured variables received from the transponder.
- Computation and storage of the current values of derived variables, which have been calculated by the server using the values of measured variables.
- Detecting and handling of alarm conditions on variables.
- Storage of historical data (trend logs).
- Presentation of the plant operation using measured as well as derived variables.
- Notification of users about alarm conditions, by means of local touch panel, e-mail messages or SMS messages.

5 Specification of SMARP Server

All types of the process data stored within the server and all types of data processing are defined in a system specification, which is written in the form of a set of XML documents. The specification is compiled by an automatic tool and deployed on the server. No custom programming is needed in order to put a server (a server instance) into the operation. This enables fast deployment and helps in keeping low design cost.

SMARP specification consists of the following documents:

- Types.xml – reusable library of types of measured and calculated variables;
- Expressions.xml – reusable library of expressions to calculate derived variables;
- Sensors.xml – definitions of measured variables;
- Derived.xml – definitions of derived variables;
- Constants.xml – definitions of constants, such as alarm thresholds;
- Alarms.xml – definitions of alarms;
- Factory.xml – description of the plant structure.

Types give semantic context to variables. A type has a name and a description of the data format, the physical entity, and the unit of measure. For example:

```
<types>
   <type name = "Temp"            type name
         data = "float"           data type, e.g. float, string etc.
         kind = "temperature"     physical entity
         unit = "C"               unit of measure, e.g. Celsius
   />
   ......................         other type definitions
</types>
```

Sensors are process variables measured by sensors attached to the plant devices and transmitted by the transponder. A variable has a name, a set of attributes and a set of values. Names and attributes are specified in the following way:

```
<sensors>
   <item name  = "t1"             variable name
         type  = "Temp"           type name
         mode  = "0"              processing mode
         cycle = "10"             measurement and reporting period
   </item>
   ......................         other variable definitions
</sensors>
```

Values of a measured as well as derived variables consist of the current value, time stamp and the current processing mode. Values are not specified in a file, however, they are stored and processed in memory during the server operation. The processing modes can be the following:

- 0 – variable is not processed;
- 1 – the value is stored in memory;
- 2 – the value is stored in memory and archived in the persistent history log.

Constants are defined as measured variables with processing mode set to 0.

Expressions define rules for calculating values of derived variables. An expression has a name, a list of formal parameters and a body written in XEXPR language developed by W3C [8]. Expressions can contain functions. For example:

```
<expressions>
   <expression name = "exp1" >      expression name
      <arguments>
         <item name = "x"           the first formal parameter
               type = "Temp"
         />
         ..................         other formal parameters
      </arguments>
      <expr>
         <add> <x/> <x/> </add>     XEXPR body (here: 2x)
      </expr>
   </expression>
   ......................           other expression definitions
</expressions>
```

Derived variables store the process data, which are not measurable directly, but which can be calculated from other variables (measurements) by means of expressions. The specification of derived variables is compatible with the specification of measured variables, but it contains a few additional attributes. For example:

```
<derived>
   <item name   = "l1"              variable name
         type   = "Unit_1"          variable type
         mode   = "0"               initial processing mode
         source = "exp2"            expression name
         order  = "120" >           sequence of computation
      <arguments>
         <item name = "t1" />       the first actual argument
         <item name = "a"  />       the second actual argument
         ..................         other actual arguments
      </arguments>
   </item>
   ........................         other variable definitions
</derived>
```

Alarms are abnormal situations, which may require intervention. Each alarm is associated with a derived variable called alarm reporting variable. If an alarm is raised, the variable is set to a positive value. Thus, an alarm is detected in the body of an

expression used to calculate value of reporting variable. Alarm has a name, a report variable, and a list of panel, sms and e-mail addresses. For example:

```
<alarms>
   <alarm name   = "A_t1"          alarm name
          report = "state" >       name of the report variable
      <phones>
         <sms>
            <to> 123456789 </to>   telephone number
            <note> text </note>    sms contents
         </sms>
         ..................        other sms definitions
      </phones>
      <mails>
         <mail>
            <to> xx@ex.com </to>   e-mail address
            <note> text </note>    e-mail contents
         </mail>
         ..................        other e-mail definitions
      </mails>
   </alarm>
   ......................          other alarm definitions
</alarms>
```

The structure of the plant equipment consists of machines, which are assembled into production lines. The structure and interconnections of those lines form a graph, which nodes are machines and edges are machine connections. Therefore, the plant structure is specified as a graph of machines, described in GraphML language [9].

6 A Model of the Server Operation

SMARP server specification can automatically be compiled into tables stored in a relational database (Fig. 3).

Attributes of all the measured and derived variables, as well as constants, are stored in a single table *Variable*. The values of those entities are stored in *Data* table. Whether *Data* is a relational table or a data structure residing in the operating memory of the server, depends on the implementation. Historical values of variables are stored in a separate *History* table.

Three tables store lists of variables. *ListS* stores derived variables that should be calculated after receiving a new value of each measured variable. *ListF* stores formal arguments of each expression. *ListA* stores actual arguments of each derived variable.

When the server is put into operation, its activities are triggered by arriving messages and by the flow of time. Each time a new transponder message arrives, the server calculates derived variables listed in table *ListS*. Panel messages are treated in the same way.

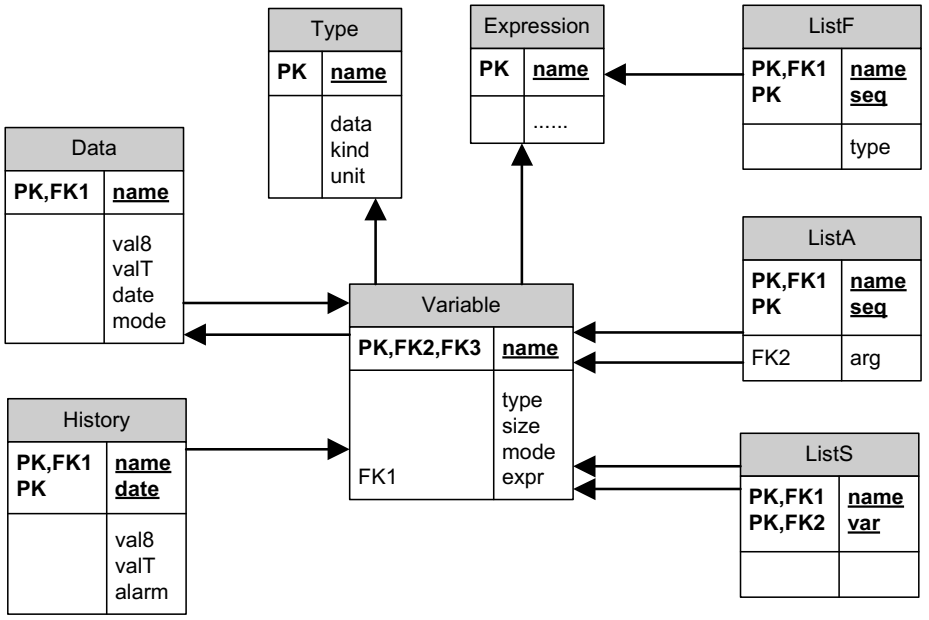


Fig. 3. Data model of SMARP server

Presentation of the plant status and operation is implemented by means of HTML web pages, which are generated automatically by the server using the specification of the plant structure and the current values of the process data stored in *Data* table.

A special mechanism for alarm handling exists in the server. The rules for raising an alarm and for handling the alarms are included into expressions that define derived variables. Each time a new value is calculated, then the alarm conditions are examined and if they are fulfilled, an alarm is raised. Alarm handling activities can display a message on a local touch panel or can send SMS or e-mail messages to the users. Displaying or sending a message is implemented by embedded server functions, which can be used within the expressions. When message is displayed on a touch panel, an operator has opportunity to enter a cause of alarm.

A lack of communication between transponder and server can be detected by the server and reported as an alarm. When communication is interrupted, there is a mechanism that allows server to order transponder to retransmit lost data.

In order to ensure that local clocks of all transponders and server are synchronized, SMARP provides a mechanism for time synchronization. It can be utilized, when there is no possibility to use standard mechanisms, such as NTP or Windows Time Service. This may happen when company policy doesn't allow for communicating external time server.

7 Conclusions and Future Work

The paper describes a new approach to production data monitoring and reporting. The monitoring and reporting system, called SMARP, is composed of a local plant transponder and a remote server located in the Internet. SMARP supports the manufacturing decision makers and provides aggregated process data that describes the effects of the plant operation, the overall equipment efficiency (OEE) and the causes of loses, such as accidents, damages and stoppages. The user can connect to the server through an arbitrary communication device, e.g. a laptop, a tablet or a mobile phone and inspect a production line status.

Currently, we are working on an application of SMARP system in an automated plant, which bottles and packs mineral water.

References

1. Boyer, S.A.: SCADA Supervisory Control and Data Acquisition. ISA - International Society of Automation, USA (2010)
2. Bailey, D., Wright, E.: Practical SCADA for Industry, Newnes (2003)
3. OPC Foundation, http://www.opcfoundation.org/Downloads.aspx
4. Ozdemir, E., Karacor, M.: Mobile phone based SCADA for industrial automation. ISA Transactions 45(1), 67–75 (2006)
5. Wallance, D.: How to put SCADA on the Internet. Control Eng. 50(9), 16–21 (2003)
6. Devices World, Malaysia, http://www.devicesworld.net/
7. Vipond Controls, Australia, http://www.vipondcontrols.ca/iscada/
8. Nicol, G.T.: XEXPR - A Scripting Language for XML, http://www.w3.org/TR/xexpr/
9. Brandes, U., Eiglsperger, M., Lerner, J.: GraphML Primer, http://graphml.graphdrawing.org/

Reliability Analysis of a Two-Stage Goel-Okumoto and Yamada S-shaped Model

Ioannis G. Sideratos, Agapios N. Platis*, Vasilis P. Koutras, and Nicholas Ampazis

University of the Aegean, Chios, Greece
platis@aegean.gr

Abstract. Software systems are becoming essential parts in many products. These are most commonly home devices and other industrial and commercial products. Nowadays, with this increasing dependence on software systems, the size and the complexity of the software products has increased dramatically. The interest on software systems has led to the development of techniques on software reliability assessment such as the Non-Homogeneous Poisson Process Models, Goel-Okumoto and Yamada S-shaped models. This paper is an analysis on the reliability of the software program of a large telecommunication company. An estimation of the Goel-Okumoto model parameters were given with the Matlab program, and the parameters of the Yamada's S-shaped model were estimated with the SMERFS software reliability tool. Additionally, we assume that new code is inserted into the software and we estimate the parameters of both models. A detailed examination of the results led us conclude that the Yamada's S-shaped model is the most appropriate model.

Keywords: NHPP models, SMERFS, software reliability measurement, Goel-Okumoto model, Yamada S-shaped model, 2-stage models.

1 Introduction

For critical business applications, high availability is a major requirement as well as the reliability which is as well an important component. Customers expect continuous availability of new software systems, but very often the software defects, is perhaps the most difficult problem, the software industry is facing today.

The pressure of the software schedule, the resource limitations and the unrealistic demands of the users can affect software reliability. The development of a reliable software system is pretty difficult when there is a dependence of the modules of the software system. It is also quite hard to know if the software that we delivered is reliable or not. After the software is marketed, its reliability is checked from the customer's feedback (problem reports, complaints or compliments etc). Suppliers of software systems need to know, if their products are reliable before they are delivered to customers. The reliability models have appeared in literature, in an attempt to provide, such information to suppliers.

* Corresponding author.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_38,

In this paper, an analysis of a particular category of time domain models, the Non-Homogeneous Poisson Process Models, was made. An analysis of the Goel-Okumoto and the Yamada S-shaped software reliability model was made and the parameters of both models had been estimated, using the failure data of a telecommunication's company software system [4]. Finally, in the last part of this paper, a reliability analysis was made [1], based on these two models, assuming that, a significant amount of code, is added on the software after a period of time.

2 Non-Homogeneous Poisson Process Models (NHPP)

A software error seems to happen when the output of the software is either different from the expected one or from the actual one. During the testing or the debugging process of the software, an error on the software is recognized after a failure and this error, that caused the failure, is removed [6]. Therefore, the same defect will not happen again. In this way, the frequency of the errors is decreasing and the software system can be released to the markets. Usually, a software reliability growth model is characterized by the frequency of its mean value function in order to analyze all the failure data. To determine the mean value function of a model, a number of parameters should be estimated from the failure data of course [4].

The Maximum Likelihood method is the most commonly used technique for estimating the parameters of a software reliability growth model. Usually, the Maximum Likelihood equations are somewhat complicated and usually solved by numerical solutions using computer programs and libraries [6]. This technique is widely used as a standard parametric technique. The NHPP models, can forecast the expected number of failures in a system of software, but also the future reliability (expressed as MTBF), using failure data from later testing phases. These models assume that, failure intensity decreases when failures and bugs in a software system are detected [4].

2.1 Goel-Okumoto Model

The first NHPP model [4, 5, 9] we introduce was presented by Amrit Goel and Kazu Okumoto on 1979. This model has the following mean value function:

$$\mu(t)= \alpha(1-e^{-bt}) , \qquad \alpha>0, b>0 \tag{1}$$

The Goel–Okumoto model is probably the most widely used software reliability model, because of its simplicity and the easy interpretation of model parameters to software-engineering-related measurements. This model, which is also a finite reliability model, assumes that there are a finite number of faults in the software and that the testing and debugging process does not introduce more faults in the software. The parameter α, stands for the cumulative number of faults eventually detected and b indicates the failure occurrence rate.

The failure intensity function of this model is the derivative of its mean value function and it is:

$$\lambda(t)= \alpha b e^{-bt} \tag{2}$$

2.2 Yamada S-shaped Software Reliability Model

The Yamada S-shaped software reliability model was proposed by Yamada and Osaki, at 1984. This model has the following mean value function [4, 7, 8]:

$$\mu(t)= \alpha(1-(1+bt)\, e^{-bt}), \qquad \alpha>0,\ b>0 \tag{3}$$

The parameter α can also be interpreted as the expected total number of faults eventually to be detected, and the parameter b represents a steady-state fault detection rate per fault. This is a finite failure model with the mean value function μ(t) showing the characteristic of an S-shaped curve rather than the exponential growth curve of the Goel–Okumoto model.This model assumes that the software fault detection process has an initial learning curve, followed by growth when testers are more familiar with the software, and then leveling off as the residual faults become more and more difficult to detect.

The failure intensity function of this model is [7]:

$$\lambda(t)=\alpha b^2\, t\, e^{-bt} \tag{4}$$

3 Implementation

The choice of the appropriate model is a very important issue in modeling and analysis of software reliability.

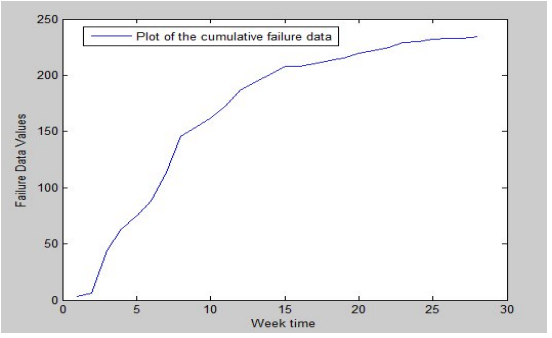


Fig. 1. A figure of the cumulative failure data vs time weeks

In Figure 1 [3], we see the shape of the cumulative failure data (Table 1) versus the time week. These data have not a concave behavior, which means that this software is not reliable, due to the fact that errors in concave models are instantly repaired [3]. So

it requires much more effort to detect the failures in the near future in this software. So probably, the Yamada's model will perform better with these failure data.

However, no single model of what we presented it has emerged as superior to the others [4]. In order to be able to successfully implement the technical reliability of software, one must choose the model that is the most suitable to all the failure data that will be analyzed.

Our failure data came from a software of a large telecommunication company:

Table 1. Data from a software of a telecommunications company ([4])

Week	Failure /cumulative		Week	Failure /cumulative		Week	Failure /cumulative		Week	Failure /cumulative	
1	3	3	**8**	32	146	**15**	7	208	**22**	3	225
2	3	6	**9**	8	154	**16**	0	208	**23**	4	229
3	38	44	**10**	8	162	**17**	2	210	**24**	1	230
4	19	63	**11**	11	173	**18**	3	213	**25**	2	232
5	12	75	**12**	14	187	**19**	2	215	**26**	1	233
6	13	88	**13**	7	194	**20**	5	220	**27**	0	233
7	26	114	**14**	7	201	**21**	2	222	**28**	1	234

3.1 Goel-Okumoto Analysis

In order to be able to find the functions of mean value and the failure intensity we will use these data for the calculation of the reliability of this system. We should first of all estimate all the parameters of the models. For the Goel–Okumoto model, the method of Maximum Likelihood Estimation is used [4]. A Matlab code was used in order to calculate the parameters of the Goel-Okumoto model. This code (available in http://fileadmin.cs.lth.se/cs/Education/ETS200/Labs/lab4_Reliability/Reliability_del/go.m) calculates the parameters α, b of this model assuming that the distance between the upper and lower bound is 0.001, with b being their average. This code also calculates the sum of the maximized log-likelihood function (partial derivative with respect to b) for b and if this sum is lower than 0, then the upper bound is equal with b, otherwise b is equal with the lower bound. The parameter α is calculated after having calculated b. The parameters of the Goel-Okumoto model are:

$$\alpha = 250.3691(\text{total number of faults}) \quad \text{and} \quad b = 0.0974\ (\text{failure rate})$$

The mean value function of Goel-Okumoto model is:

$$\mu(t) = 250.3691(1-e^{-0.0974t})$$

The mean value function of the Goel-Okumoto model is shown in Figure 2:

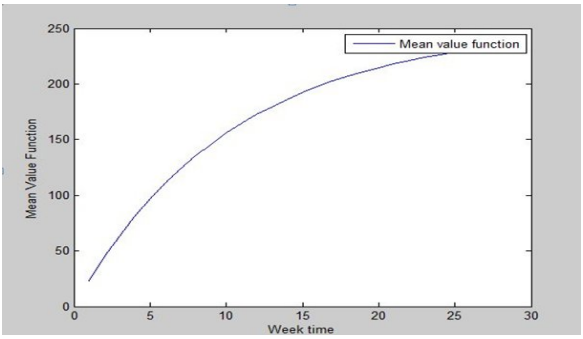


Fig. 2. Mean value function of the Goel-Okumoto model

3.2 Yamada S-shaped Analysis

For the Yamada's S-shaped software reliability model, a reliability tool is used that was developed from Dr. William Farr [2], the SMERFS (Statistical Modeling and Estimation of Software Reliability Functions) tool. This a program for the estimation and predictions of the model parameters and the reliability of software systems during the testing phase. This tool allows the user to select a specific model but does not interpret the results which it exports. Analysts should understand the results. This tool calculates all the mathematics and statistics that are needed to interpret all the failure data. This reliability tool, also gives all the parametric values of the models but the analyst himself, is the one who should decide what results of them are meaningful.

This tool [2] also produces diagrams for each model with the estimate of all total failures but also of the remaining faults. The user is asked to specify whether the data is time-between–failure or interval data.

So we give the SMERFS tool the failure data of the software (Table 1) and we have the following estimations for the Yamada's model:

Table 2. Estimation of S-shaped model using the SMERFS reliability tool.

Yamada's S-shaped model	α= 236.62 (total faults by this model)	b= 0.2326 (failure occurrence rate)

The mean value function of the S-shaped reliability model is shown in the following figure:

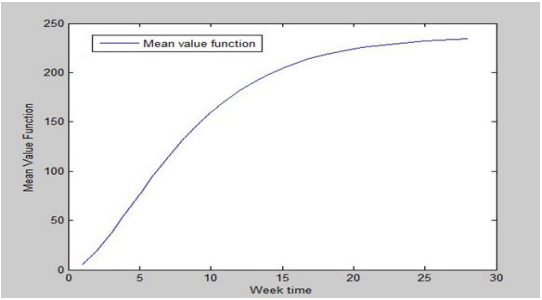


Fig. 3. Mean value function of the Yamada's S-shaped model.

The failure intensity of this model (SMERFS tool produces it) is shown in the following figure:

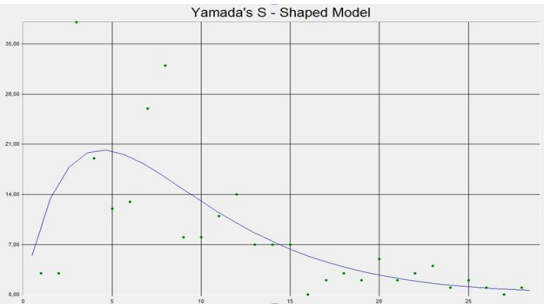


Fig. 4. Failure intensity function of Yamada's S-shaped model by SMERFS reliability tool

3.3 Two-Stage Goel-Okumoto and Yamada S-shaped Models

Finally, in our last part of this paper, the reliability analysis is presented, based on the failure data, assuming that a large amount of a new code is added in this software [1]. So it's like having two software systems with the following, Goel-Okumoto, for example, overall mean value function:

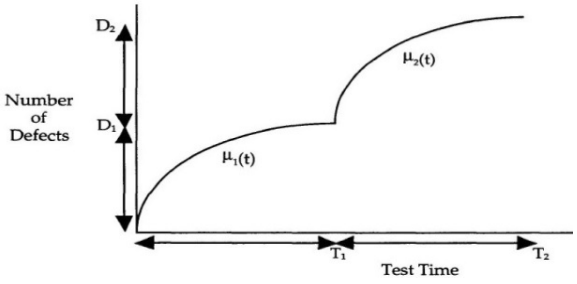


Fig. 5. 2- staged Goel-Okumoto model ([1])

It should be noted, that this analysis is also carried out, with the model of Yamada and Osaki (S-shaped model). So here we have the failure data for the two software systems:

Table 3. Data from a software until the 14th week ([4]).

Week	Failure / Cumulative		Week	Failure / Cumulative	
1	3	3	**8**	32	146
2	3	6	**9**	8	154
3	38	44	**10**	8	162
4	19	63	**11**	11	173
5	12	75	**12**	14	187
6	13	88	**13**	7	194
7	26	114	**14**	7	201

Table 4. Data from a software till 28th week

Week	Failure / Cumulative		Week	Failure / Cumulative	
15	10	10	**22**	35	170
16	3	13	**23**	12	182
17	40	53	**24**	9	191
18	22	75	**25**	13	204
19	14	89	**26**	15	219
20	18	107	**27**	7	226
21	28	135	**28**	8	234

So the failure data are on the two above tables. Let's assume that we have two Goel-Okumoto models. It is also assumed [1], that the first mean value function is for the period from the 1st week to the 14th week and the second mean value function is for the period of the 15th week to the 28th week. For the first Goel-Okumoto model, the Maximum Likelihood techniques are used, as we have done on the previous part of this paper in order to estimate the parameter values of this model. Thus, we found that:

$$\underline{\alpha_1 = 563.7287} \qquad \underline{b=0.0315}$$

The mean value function of the first Goel-Okumoto model is shown in the following figure:

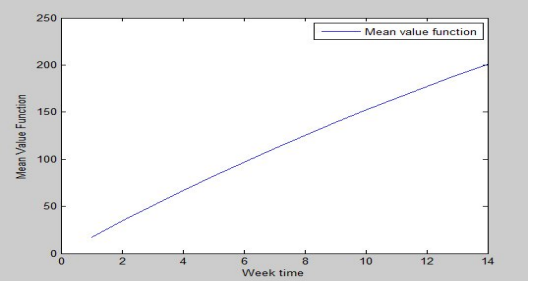


Fig. 6. Mean value function of the first Goel-Okumoto model

And for the second Goel-Okumoto model the parameters and the mean value function are:

$$\underline{\alpha_2 = 542.8589} \quad \text{and} \quad \underline{b= 0.0403}$$

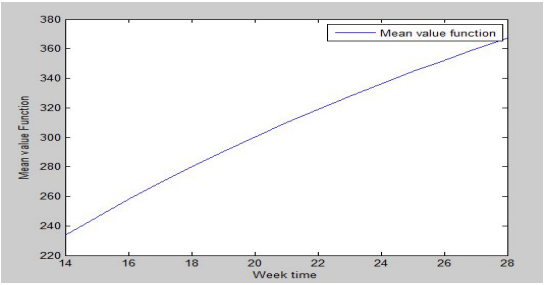


Fig. 7. Mean value function of the second Goel-Okumoto

And for both Goel-Okumoto models, the overall mean value function is something like this:

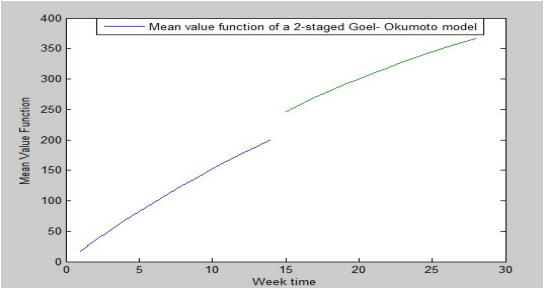


Fig. 8. Mean value function of the 2-staged Goel-Okumoto software reliability model

For the Yamada's S-shaped reliability model [1], we still have the same failure data. So, the SMERFS reliability tool is used to calculate the parameters of the S-shaped model. So, in the first case, for the first Yamada's model, (1st week- 14th week) we have:

$$\alpha_1 = 240.1280 \quad \text{and} \quad b = 0.2331$$

And for the second case (15th week – 28th week):

$$\alpha_2 = 273.277 \quad \text{and} \quad b = 0.2448$$

So, the overall mean value function for both the Yamada's S-shaped models is:

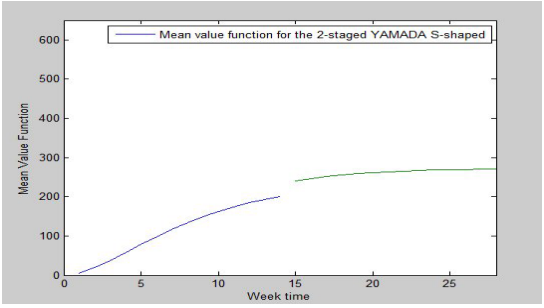


Fig. 9. Mean value function of the 2-staged Yamada's S-shaped reliability model

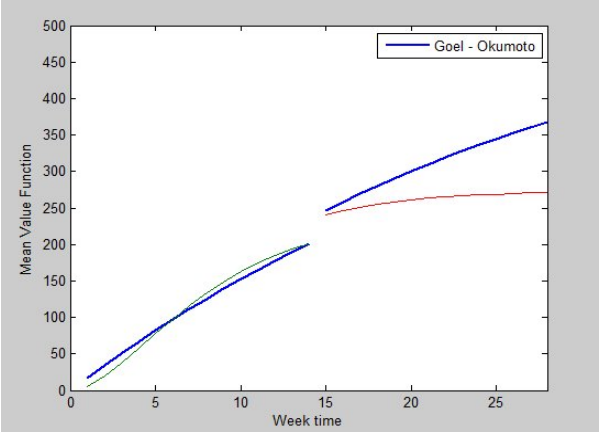


Fig. 10. Overall comparison figure of the two models (Goel-Okumoto and Yamada S-shaped)

In this last figure, where the "thicker" (blue) curves are the curves of the Goel-Okumoto model, a drop (fall) of the second Yamada's model (red curve) is observed and greatly strengthens our view, that the model of Yamada is the best for these specific failure data.

4 Conclusions

In order to implement a model of software reliability we need to estimate the parameters of this model. These estimates are based on the failure data that we have chosen. With the Maximum Likelihood Estimation method the parameters of Goel-Okumoto model are estimated for the failure data of a software system from a large telecommunications company and the parameters of the Yamada's S-shaped software reliability model are estimated with the SMERFS reliability tool. Based on these estimates for the two models, we see that we have fewer errors in Yamada's S-shaped model (around 6%) than the Goel-Okumoto model. The cumulative (real) failures [3] in the table tend to create an 'S' (small values of failure data in the beginning of the test, then quite larger and very small values at the end). For this reason, it is not surprising that the model of Yamada (S-shaped) behaves better than the Goel-Okumoto model. Furthermore, the values of failure data in the failure intensity function of Yamada's model "fit" the curve, so it is quite reasonable, the fact that Yamada's S-shaped model behaves much better than the Goel-Okumoto model [3].

In the last part of this paper, it is assumed that new code is inserted to the software [1]. So we keep the same failure data until the 14^{th} week, while right after this time period, the failures of the 15^{th} week onwards (Table 1), are added to the data of the previous period. It's like having two different software systems, one buggy enough and one with much more errors. Having estimated the parameters of both models we found that once again, S-shaped model behaves better than the Goel-Okumoto model, as there seems to be a reduction of errors in both cases. In the case of this test, until

the 14th week we have a reduction of about 57%, while after the addition of the new code we have a decrease of 49.6% in the same model.

References

1. Wood, A.: Software Reliability Growth Models. Technical Report 96.1 (September 1996)
2. Wallace, D.R.: Practical Software Reliability Modeling. Software Assurance Technology Center, Greenbelt (2001)
3. Ramos, I.C.: Statistical procedures for Certification of software systems, Available from the Eindhoven University of Technology, Sevilla Spain (2009)
4. Xie, M., Hong, G.Y., Wohlin, C.: Modeling and Analysis of Software System Reliability. In: Blischke, W., Murthy, P. (eds.) Case Studies on Reliability and Maintenance. Wiley VHC Verlag, Germany (2003)
5. Barraza, N.R.: Parameter Estimation for the Compound Poisson Software Reliability model, School of Engineering, University of Buenos Aires, Argentina
6. Lai, R., Garg, M.: A Detailed Study of NHPP software reliability models, Department of Computer Science, La Trobe University, Victoria, Australia (June 2012)
7. Charles Ilayaraja, S., Ganaga Durga, M.: A Comparative Study of Failure Data for Software Reliability Estimation (April 2013)
8. Gokhale, S.S., Marinos, P.N., Trivedi, K.S.: Important Milestones in Software reliability modeling. Department of Electrical and Computer Engineering, Duke University, Durham (1996)
9. Almering, V., van Genuchten, M., Cloudt, G., Sonnemans, P.: Using software reliability growth models in practice. IEEE Software (2010)

Reliability Assessment of Cooperation and Replacement of Surveillance Systems in Air Traffic

Mirosław Siergiejczyk, Karolina Krzykowska, and Adam Rosiński

Warsaw University of Technology, Faculty of Transport
00-662 Warsaw, Koszykowa 75
{msi,kkrzykowska,adro}@wt.pw.edu.pl

Abstract. In Poland, a problem of lack of radar coverage at certain flight levels over certain regions of the country is being observed. Therefore, the Polish Air Navigation Services Agency is looking for some better surveillance solutions for airspace and airport surface. Multilateration system (MLAT) are part of the technology used in aviation for many years. Originally developed for military purposes - to identify and determine the position of military airplanes in the air. In the system TDOA location method was used (Time Difference of Arrival) - the difference between flight time of the aircraft over the sensors.

Economic analyzes have shown that the installation and maintenance of the MLAT, is cheaper and provide more benefits than installing an additional SSR radar. The MLAT can provide the desired coverage area, and most importantly - can more quickly increase the accuracy of navigation of the aircraft. In addition, it is worth noting that no moving part of MLAT system is its additional advantage. The paper presents an analysis of surveillance systems with particular emphasis on the of the possibility of replacing radar systems with multilateration. Improving the reliability of the system utility can be done by reducing the recovery time of suitability.

Keywords: surveillance, reliability, air traffic.

1 Introduction

One of the major problems of air traffic and transport in Poland is quantitative and qualitative development of technical communication, navigation and surveillance systems. The point is that the authorities providers of navigation and surveillance services ensured technical security of the movement. It is important to take into account the implementation of European air traffic management programs, including programs for implementing the Single European Sky (SES) - such as SESAR (SingleEuropeanSky ATM Research). [18]

In Poland, a problem of lack of radar coverage at certain flight levels over certain regions of the country is being observed. Therefore, the Polish Air Navigation Services Agency is looking for some better surveillance solutions for airspace and airport surface.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_39,

Radar techniques were presented for the first time in the thirties of the twentieth century. The first radar equipment came from radio equipment HF and VHF range. Radiolocation, as the branch of radio, involves using electromagnetic waves to detect objects in space, as well as to determine the location and motion parameters. World War II was a period of unusually expansive development of radar, contributing to the success of many important military operations. After the war, this area was already fully educated sphere of technology. Another important stage in the development of radar breakthrough occurred in the fifties and sixties of the twentieth century. As an achievement of this period may be mentioned, among others, development of methods for coding and pulse compression, noise reduction, and electronic search of space. In aviation, the creation and development of radar is closely linked to the needs of ground traffic control, which with a picture of the situation in the air provide navigation guidance to the pilot by radio . There are two types of radar because of the way of action. There are active and passive radars. Active works by sending radio waves into the localized object, and then waiting for a response and received it in the form of, for example, the reflected echoes of the radio. Passive applies only to reception of electromagnetic waves caused by radiation own object detected. Distance from the radar equipment is determined by the measurement of time traveling away from the radar to the detected object and back.

In the aviation technique pulse radars prevail. These radars works in a rhythm, which is responsible for the timing generator. A crucial element in the radar is antenna. It is often the most expensive element, and therefore it is important to pay a lot of attention to its parameters. [1,6] Nowadays, we have primary and secondary surveillance radars.

2 Secondary Surveillance Radars (SSR)

Secondary radar sees only aircraft equipped with a transponder, which is a wireless communication device, allowing identifies the aircraft and determines its height. The principle of SSR radar is sending a pulse train at a given frequency (1030 MHz) and waiting for a response. The transponder responds at 1090 MHz, if specifies the resulting pulse train as normal. This means that the transponder is a radio transmitter and receiver, which operates at a frequency of the radar. The diagram below illustrates the principle of SSR (fig. 1).

Transmission between the transponder and the secondary radar is at one mode. The mode we call the interrogator pulse sequence which extractor uses to determine the parameters of the position of the aircraft. There are the following modes used nowadays:

• Mode A - mode identifying civilian aircraft in the approach landing path, the correlation of the data received from flight plans allows to label the landing aircraft; transmission in A, however, is often unreliable;

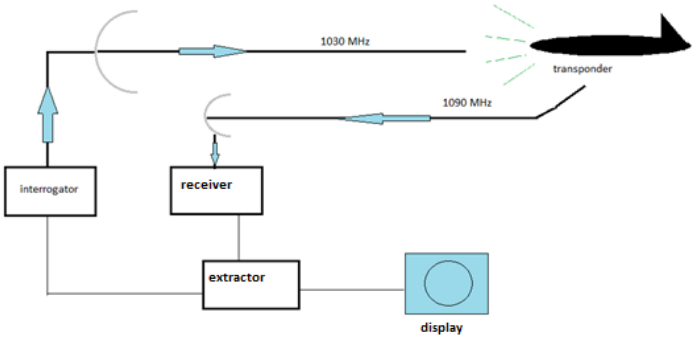


Fig. 1. Architecture of SSR radar

• Mode 3 (3 / A) - used in military aviation, on request in this mode - the transponder corresponds to the four-digit code assigned by air traffic control;

• Mode C - transponder replies stating the flight level encoding altimeter set to QNE pressure;

• Mod S - gives you the ability to assign individual 24 -bit ICAO code for each aircraft, for a total of 16 777 216 possible codes, may use bilateral data link (fig. 2).

Similarly to the primary radar, secondary radar two -flying aircraft close to each other can imagine on the display as a single marker. With heavy traffic near airports, the phenomenon of garbling is very possible. The only solution to this is a selective number, so that only one answer is requested at a time. This allows the operation of the system using mode S (selective). In the case of S, each member of the ICAO had been assigned a block of codes that can be assigned to aircraft. The way to distinguish

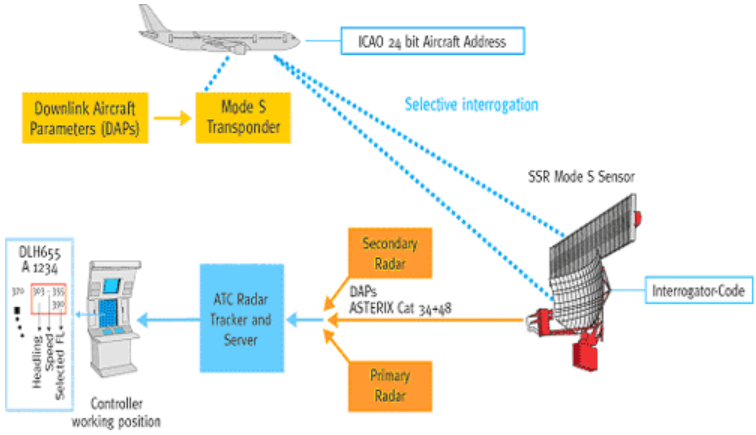


Fig. 2. SSR radar work with mode S

broadcast codes of civil aircraft depends solely on military encoding rules prevailing in the Member State of ICAO. If necessary, the codes can also be transmitted to other vehicles traveling on the airport surface. [11]

Application of SSR radar enables the identification of aircraft using a variety of procedures, such as:

- Recognition of the distinctive emblem of the aircraft on radar label;
- Recognition of the radar label previously assigned a unique code;
- Recognition of the radar label distinctive emblem of the aircraft equipped with Mode S;
- Transfer of radar identification;
- Observing the command to adjust specific code.

3 Multilateration System (MLAT)

Multilateration system (MLAT) are part of the technology used in aviation for many years. Originally developed for military purposes - to identify and determine the position of military airplanes in the air. In the system TDOA location method was used (Time Difference of Arrival) - the difference between flight time of the aircraft over the sensors. Sensors are one of the basic components of the MLAT. Their strategic location should allow the greatest possible coverage of airspace. Its task is to listen for answers to questions typical for secondary surveillance radar. Each station receives the response of the aircraft at another time. With advanced computing techniques hyperboloids are determined for each sensor. The intersection of the hyperboloid is the identification of the position of the aircraft. To locate an object in 3D space requires at least four antenna feeders, which results rule that the N antennas of the receiver allows you to specify N - 1 hyperboloid (fig. 3).

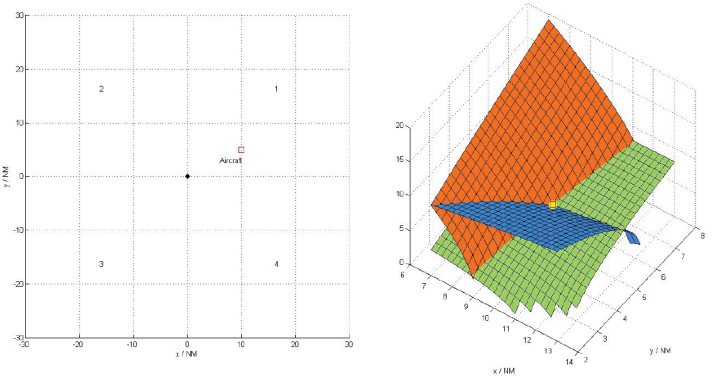


Fig. 3. Multilateration system workout

Economic analyzes have shown that the installation and maintenance of the MLAT, is cheaper and provide more benefits than installing an additional SSR radar. The MLAT can provide the desired coverage area, and most importantly - can more quickly increase the accuracy of navigation of the aircraft. In addition, it is worth noting that no moving part of MLAT system is its additional advantage. The sensors used are small and easily accessible, i.e. the radar needs all infrastructure and massive current source and a sensor can be easily located on an existing structural member. Comparison of the two systems architecture is shown in Fig.4.

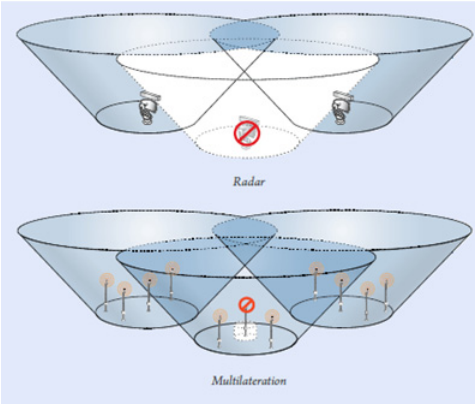


Fig. 4. Comparison of the two systems architecture

However, an important advantage of the hyperbolic system is the fact that the exclusion of one sensor would not exclude from working the entire system. The required coverage area is achieved by other operating elements. Interruption in power supply to one of the radar causes no cover to a large part what may constitute a danger in performance of various air operations. [16]

4 Analysis of the Possibility of Replacing Radar Systems with Multilateration in Terms of Reliability

It is possible to create an advanced integrated system which consists of different sources of surveillance, for example multilateration and radar, where we would have basic system (multilateration) and reserve system(radar).

Damage of one of the systems moves it from the state of full ability R_O (t) to the state of the impendency over safety unreliability Q_{ZB} (t) [12,13,14]. Figure 5 shows the relationships in the integrated surveillance system in terms of reliability [15].

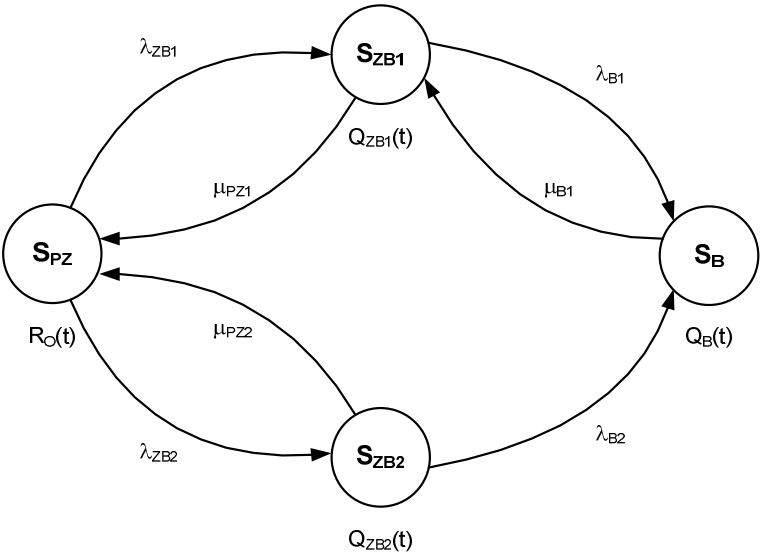


Fig. 5. Relations in the system

Denotations in figures:

$R_O(t)$ – the function of probability of system staying in state of full operational capability

$Q_{ZB}(t)$ – the function of probability of system staying in state of the impendency over safety,

$Q_B(t)$ – the function of probability of system staying in state of unreliability of safety,

λ_{ZB1}, λ_{ZB2} – transition rate from the state of full ability into the state of the impendency over safety,

μ_{PZ1}, μ_{PZ2} – transition rate from the state of the impendency over safety into the state of full ability,

λ_{B1}, λ_{B2} – transition rate from the state of state of the impendency over safety into the state of unreliability of safety.

The system illustrated in fig. 5 may be described by the following Chapman–Kolmogorov equations:

$$\begin{aligned}
R_0'(t) &= -\lambda_{ZB1} \cdot R_0(t) + \mu_{PZ1} \cdot Q_{ZB1}(t) - \lambda_{ZB2} \cdot R_0(t) + \mu_{PZ2} \cdot Q_{ZB2}(t) \\
Q_{ZB1}'(t) &= \lambda_{ZB1} \cdot R_0(t) - \mu_{PZ1} \cdot Q_{ZB1}(t) - \lambda_{B1} \cdot Q_{ZB1}(t) + \mu_{B1} \cdot Q_B(t) \\
Q_{ZB2}'(t) &= \lambda_{ZB2} \cdot R_0(t) - \mu_{PZ2} \cdot Q_{ZB2}(t) - \lambda_{B2} \cdot Q_{ZB2}(t) \\
Q_B'(t) &= \lambda_{B1} \cdot Q_{ZB1}(t) + \lambda_{B2} \cdot Q_{ZB2}(t) - \mu_{B1} \cdot Q_B(t)
\end{aligned} \tag{1}$$

Given the initial conditions:

$$\begin{aligned} R_0(0) &= 1 \\ Q_{ZB1}(0) &= Q_{ZB2}(0) = Q_B(0) = 0 \end{aligned} \tag{2}$$

The following system of linear equations we get after Laplace transform:

$$\begin{aligned} s \cdot R_0^*(s) - 1 &= -\lambda_{ZB1} \cdot R_0^*(s) + \mu_{PZ1} \cdot Q_{ZB1}^*(s) - \lambda_{ZB2} \cdot R_0^*(s) + \mu_{PZ2} \cdot Q_{ZB2}^*(s) \\ s \cdot Q_{ZB1}^*(s) &= \lambda_{ZB1} \cdot R_0^*(s) - \mu_{PZ1} \cdot Q_{ZB1}^*(s) - \lambda_{B1} \cdot Q_{ZB1}^*(s) + \mu_{B1} \cdot Q_B^*(s) \\ s \cdot Q_{ZB2}^*(s) &= \lambda_{ZB2} \cdot R_0^*(s) - \mu_{PZ2} \cdot Q_{ZB2}^*(s) - \lambda_{B2} \cdot Q_{ZB2}^*(s) \\ s \cdot Q_B^*(s) &= \lambda_{B1} \cdot Q_{ZB1}^*(s) + \lambda_{B2} \cdot Q_{ZB2}^*(s) - \mu_{B1} \cdot Q_B^*(s) \end{aligned} \tag{3}$$

Modelling of Process of System Maintenance

Computer simulation and computer-aided analysis facilitate to relatively quickly determine the influence of change in reliability-exploitation parameters of individual components on reliability of the entire system. Of course, the reliability structure of both the entire system and its components has to be known beforehand.

Using computer aided allows to perform the calculation of the value of probability of system staying in state of full operational capability R_O. That procedure is illustrated with below example.

Example:

The following quantities were defined for the system:

-test duration - 1 year (values of this parameter is given in [h]):

$$t = 8760 \ [h]$$

-transition rate from the state of full ability into the state of the impendency over safety λ_{ZB1} (failure of multilateration):

$$\lambda_{ZB1} = 0{,}000006$$

-transition rate from the state of full ability into the state of the impendency over safety λ_{ZB2} (failure of radar):

$$\lambda_{ZB2} = 0{,}000001$$

-transition rate from the state of state of the impendency over safety into the state of unreliability of safety λ_{B1} (failure of multilateration):

$$\lambda_{B1} = 0{,}000001$$

-transition rate from the state of state of the impendency over safety into the state of unreliability of safety λ_{B2} (failure of radar):

$$\lambda_{B2} = 0{,}000006$$

- intensity transition rate from the state of the unreliability of safety into the state of the impendency over safety μ_{B1}:

$$\mu_{B1} = 0{,}1$$

We obtain:

$$R_0^*(s) = \frac{\begin{array}{l} 3{,}00003\cdot 10^{11}\cdot s + 3\cdot 10^{11}\cdot \mu_{PZ1} + 5\cdot 10^{17}\cdot s^2\cdot \mu_{PZ1} + 5\cdot 10^{17}\cdot s^2\cdot \mu_{PZ2} + 5{,}00035\cdot 10^{16}\cdot s^2 + \\ + 5\cdot 10^{17}\cdot s^3 + 5{,}0003\cdot 10^{16}\cdot s\cdot \mu_{PZ1} + 5{,}00005\cdot 10^{16}\cdot s\cdot \mu_{PZ2} + 5\cdot 10^{16}\cdot \mu_{PZ1}\cdot \mu_{PZ2} + \\ + 5\cdot 10^{17}\cdot s\cdot \mu_{PZ1}\cdot \mu_{PZ2} \end{array}}{\begin{array}{l} 2{,}100021\cdot 10^{6}\cdot s + 5{,}00035\cdot 10^{16}\cdot s^2\cdot \mu_{PZ1} + 5{,}00035\cdot 10^{16}\cdot s^2\cdot \mu_{PZ2} + 5\cdot 10^{17}\cdot s^3\cdot \mu_{PZ1} + \\ + 5\cdot 10^{17}\cdot s^3\cdot \mu_{PZ2} + 6{,}500275\cdot 10^{11}\cdot s^2 + 5{,}0007\cdot 10^{16}\cdot s^3 + 5\cdot 10^{17}\cdot s^4 + 3{,}50003\cdot 10^{11}\cdot s\cdot \mu_{PZ1} + \\ + 3{,}00003\cdot 10^{11}\cdot s\cdot \mu_{PZ2} + 5\cdot 10^{16}\cdot s\cdot \mu_{PZ1}\cdot \mu_{PZ2} + 5\cdot 10^{17}\cdot s^2\cdot \mu_{PZ1}\cdot \mu_{PZ2} \end{array}}$$

Assuming $\mu_{PZ1} = 0{,}1$, $\mu_{PZ2} = 0{,}2$ and using the Laplace'a transformation we receive:

$$R_O = 0{,}999935$$

The data contained in the example are given for instance in order to verify the correctness of the methodology presented. In further consideration of the described system the authors expect to carry out research in selected airports and obtain real data on the analyzed systems.

5 Summary

The presented process of surveillance systems analysis allows us to determine the level of reliability of the proposed integrated system. This is possible by using the different two systems which can work together and that can ensure an adequate level of reliability indicators.

The paper presents an analysis of surveillance systems with particular emphasis on the of the possibility of replacing radar systems with multilateration. Improving the reliability of the system utility can be done by reducing the recovery time of suitability. In a further study of this issue should be sought to determine the relationship between financial inputs, sometimes associated with the restoration of full ability, and the probability of systems staying in the highlighted technical conditions.

References

1. Bajda, A., Wrażeń, M., Laskowski, D.: Diagnostyka jakości transferu danych w procesie zarządzania sytuacją kryzysową. Electrical Review 9a'2011, Sigma-Not, Warszawa (2011)
2. Domicz, J., Szutowski, L.: Podręcznik pilota samolotowego. Poznań (1998)

3. Januszewski, J.: Systemy satelitarne GPS Galileo i inne. Warszawa (2007)
4. European Organisation for the Safety of Air Navigation EUROCONTROL, NAV-GNSS Global Navigation Satellite System Training Provided by the IANS ATM Unit. Luxembourg, Kirchberg (2007)
5. ESA, European Geostationary Navigation Overlay Service (2012), http://www.cbk.waw.pl (accessed December 27, 2012)
6. Lubkowski, P., Laskowski, D.: The end-to-end rate adaptation application for real-time video monitoring. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 295–305. Springer, Heidelberg (2013)
7. Epstein, B., Weissman, I.: Mathematical models for systems reliability. CRC Press / Taylor & Francis Group (2008)
8. Kołowrocki, K., Soszyńska-Budny, J.: Reliability and safety of complex technical systems and processes. Springer, London (2011)
9. Multilateration Executive Reference Guide, Creativerge (2007)
10. Procedury Służb Żeglugi Powietrznej. Zarządzanie Ruchem Lotniczym (PL – 4444). International Civil Aviation Organization, Montreal (1996)
11. Principles of Mode S Operation and Interrogator Codes, European Organisation for The Safety of Air Navigation, Brussels (2003)
12. Rosiński, A.: Reliability analysis of the electronic protection systems with mixed – three branches reliability structure. In: Proc. International Conference European Safety and Reliability (ESREL 2009), Prague, Czech Republic, pp. 1637–1641 (2009)
13. Rosiński, A.: Reliability analysis of the electronic protection systems with mixed m–branches reliability structure. In: Proc. International Conference European Safety and Reliability (ESREL 2011), Troyes, France (2011)
14. Ważyńska-Fiok, K., Jaźwiński, J.: Reliability of technical systems. PWN, Warsaw (1990)
15. Rosiński, A.: Design of the electronic protection systems with utilization of the method of analysis of reliability structures. In: Proc. Nineteenth International Conference On Systems Engineering (ICSEng 2008), Las Vegas, USA, pp. 421–426 (2008)
16. Siergiejczyk, M., Rosiński, A., Krzykowska, K.: Reliability assessment of supporting satellite system EGNOS. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 353–363. Springer, Heidelberg (2013)
17. Siergiejczyk, M.: Maintenance Effectiveness of Transport Telematics Systems. Scientific Works of the Warsaw University of Technology, Transport Series, No. 67, Warsaw (2009)
18. Skorupski, J., Malarski, M., Stelmach, A.: Air traffic safety investigation problem. In: Proceedings and Monograph in Engineering, Water and Earth Sciences, ESREL 2006, Portugal (2006)

Swarm Intelligence Metaheurisics Application in the Diagnosis of Transformer Oil

Anis Smara, M 'hana Bouktit, and Ahmed Boubakeur

LRE / HV Laboratory, Ecole Nationale Polytechnique . Avenue Pasteur, Hacen Badi, B.P.182, El-Harrach, 16200 Algiers, ALGERIA
{Anisenp,mhana.bouktit}@gmail.com
ahmed.boubakeur@g.enp.edu.dz

Abstract . The monitoring of transformers ensures the continuity of their operation. The diagnosis of the insulating oil is an effective method of preventing malfunctions and avoids them. This work involves the adaptation of two swarm intelligence metaheuristics: ant colony optimization and bee colony optimization in the diagnosis of transformer oil. A method of hybridization of the two approaches is proposed. The comparison of the three algorithms shows that the hybrid algorithm yields better results.

1 Introduction

Electrical transformers play a major role in the distribution and the transmission of electrical energy, and we need a reliable diagnosis to avoid accident and maintain the service [1]. Mineral oil used in power transformers reveals the state of the active part. [2] – [3]. The use of the methaheuriscs is a promoted technique in the domain of diagnosis [4].In this chapter we will elaborate the diagnosis of the transformer's oil by the ant colony optimization, the bee colony optimization and we will propose a hybrid structure. [5]

The ant colony optimization is designed to solve combinatorial problems, such as the problem of finding the shortest path (the salesman problem). [6] – [7]

The bee colony optimization is also a multi-agent system that explores the space solution, and provides us with what we call feasible solutions. Both algorithms use different strategies of search, and have different features.

2 A General Approach to Solve the Problem of Diagnosis

Formally, a problem of combinatorial optimization is triple (S, f, Ω), where S is the set of candidate solutions, f is the objective function that assigns a value to each solution « s » belonging to the set S, and Ω is the set of constraints. The set of solutions « $\overline{S} \subseteq S$ » That satisfies the constraints « Ω » is called the feasible solutions.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_40,

Our main work is to adapt the problem of transformer's oil diagnosis to the ACO and the BCO algorithm.

Considering the nature of problems solved by these two algorithms, the transformer's oil problem needs to be formulated in term of a combinatorial problem.

We acquired a data base gathering the physico-chemical tests results for a large number of samples, which are regrouped into four categories (to keep, to filter, to regenerate, to discard).

The main idea is to compare the tested sample with the samples of the data base. The decision attributed to the tested sample will be the same as the decision on the most similar sample of the data base.

To get more reliable decision, the tested sample is compared with more than one sample and the decision will be represented as percentages of similarity.

Several functions of comparison can be considered. The simplest one is expressed as

$$F(E,E_b)=\frac{1}{6}[\sum_{i=1}^{6}|C_E(i)-C_b(i)\ |] \tag{1}$$

CE The six characteristics of the oil to test.

CB The characteristic of the sample picked by the ants from the database.

The function of comparison is used to calculate the heuristic information and the amount of pheromone.

2.1 The Law of Movement

During the construction of the tour, the ants move from one sample to another according to a law of movement defined by a probability and obey to a predefined constraint.

$$p_{ij}^{k}(t)=\frac{\tau_j(t)^{\alpha}\eta_{ij}^{\beta}}{\sum_{l\notin J_i^k}\tau_j(t)^{\alpha}\eta_{ij}^{\beta}} \quad \text{If} \quad j\notin J_i^k$$

$$p_{ij}^{k}(t)=0 \quad \text{If} \quad j \in J_i^k \tag{2}$$

J_i^k J_i^k : The set of the visited samples, and « t » is the iteration.

The constraint imposes a zero probability for the visited samples.

2.2 The Heuristic Information

The heuristic information is the desire to move from sample 'I' to a sample 'j', it is defined using the function of comparison F. At first, the movement of ants starts from random samples, and then it converges eventually to a tour in which the components are of the same decision. The heuristic information optimizes the time of computation by guiding ants to start evaluating samples with similar decision first.

2.3 The Pheromone

The pheromone is a principle parameter that determines the performance of our algorithm. The amount of pheromone can be constant or variable. In order to reduce the time of computation, the amount of pheromone is variable as a function of the quality of the solution.[3.2]

$$\Delta\tau(tour) = \frac{1}{\sum_{i=1}^{tr} F(E, E_i)} \tag{3}$$

$$\tau_j(t+1) = \tau_j(t) + \Delta\tau(tour) \tag{4}$$

E: The tested sample.
Ei: the sample « i » of the formed tour (the solution)
tr: The number of samples that form the tour
τ_j : The pheromone of the sample « j ».
$\Delta\tau$: The amount of pheromone
When an ant completes a tour, it starts the next from the sample at which it had stopped.

2.4 The Evaporation

The evaporation encourages the ants to explore new paths and allows them to forget the bad solutions, thus it limits the search in a sub-set of optimal solutions.

$$\tau(t+1) = \tau_j(t) \times (1-\rho) \quad (5)$$

τ_j : The pheromone of the sample « j ».

Where ρ is a real number from [0, 1] defining the rate of evaporation. [8]

According to the experts, the sample 1 is to keep, the sample 2 is to regenerate, the sample 3 to filter and the sample 4 is to throw.

These graphs represent the decision, optimized by the ant colony in percentage, as a function of the number of iterations.

We apply the ACO, but only the results on the sample two from Table 1 are plotted since it sheds light on all the problems we had so far.

Table 1. The fours samples analyzed by the experts

The samples	*The color index*	the viscosity at 40°C (Cts)	*The acidity* (mgKOH/g)	Dielectric Strength (kV/cm)	The Dissipation factor	The water content (p.p.m)
1	0.7	10.23	0.012	57	0.072	16
2	2.3	10.87	0.091	22	0.019	40
3	4	10.23	0.06	32	0.063	26
4	4.5	11.19	0.42	30	0.55	42

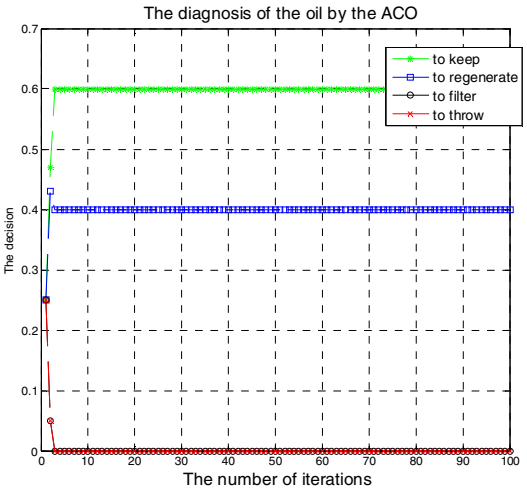


Fig. 1. The diagnosis of the sample E2 by the ACO

The decision given by the ACO for samples « 1 », « 2 », « 3 » are fine, however the decision for sample « 2 » (which is to keep) does not agree with the one provided by the experts (to regenerate). We notice that the ants converge to the solution after only a few iterations, Fig. 1; this is due to the accumulation of pheromone by the first samples trapping the ants inside a loop.

To eliminate this problem, we introduce the algorithm of the roulette wheel selection.

The origin of the algorithm is a randomness game known in the casino. The algorithm is designed such as the randomness is present; but the best elements have the highest chances to be chosen. [9]

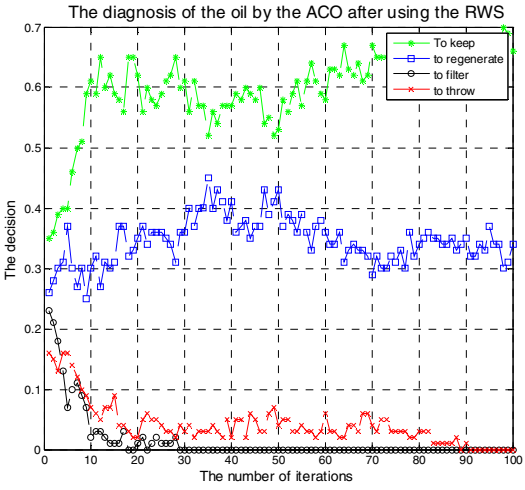


Fig. 2. The diagnosis of the sample E2 by the ACO+ RWS

Using the function of comparison we attribute weights to the samples upon their qualities defined by the pheromone and the heuristic information. The ants use the RWS on the sample's weights to choose the next sample to visit.

The application of the RWS algorithm helps the ants to avoid getting into a loop.

We also note that all the decisions were up to the requirements of the experts except for the sample « 2», which is "regenerate" and not "to keep" as yielded by the ACO fig. 2.

In fact, the regeneration is a procedure that eliminates, by use of Chemicals and adsorbents, contaminants and acid colloidal and degradation products of the oil, whereby the property that has an influence on the decision regenerate is the acidity of the oil, otherwise the oil is likely to keep. We note that the function of comparison considers the information given by the value of the acidity as one sixth of that provided by the other characteristics. This behavior neglects the information given by the acidity and lead to false results.

To mend to this problem, the artificial ants must check the acidity when they start converging to the decision "to keep" and decide whether the sample is in fact "to keep" or "to regenerate". This way the decision does agree with that of experts Fig. 3.

The decision "Filter" depends mainly on the values of the dielectric strength, the water content and the loss factor "tan δ", since the latter will be improved to the extent that it depends on the water content.

The same procedure is applied to consider the information given by the values of these proprieties in the computation of the function of comparison.

The ACO algorithm has proved its ability to find a solution; nevertheless, it is possible to improve its performance by studying the influences of its parameters on the convergence and the computing time.

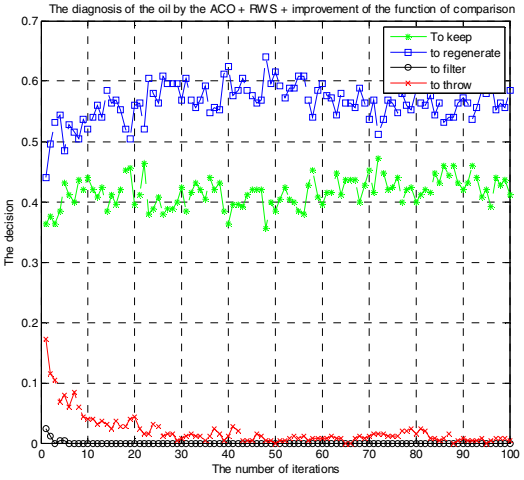


Fig. 3. The diagnosis of the sample E2 by the ACO after the improvement

Looking at the strategy used by the ACO we can say that it performs a search in the set of complete solution. This gives us an idea about the hybridization.

3 The BCO Algorithm

In order to understand the approach of the problem in the case of BCO, we propose to find the p-points centered on a predefined point, that have the smallest path. [10] – [11].

The points represent the samples of the data base, and the center point represents the tested sample.

Technically it's the same approach of the ACO but the strategy of search differs.

At first, each bee generates a partial solution of « m » components. These components are chosen according to the function of comparison between the tested sample (the center) and the samples of the data base (the other points), that we apply to RWS.

After the construction of all the partial solutions, the bees return to the hive to evaluate and communicate information about its solutions, the function of evaluation will be the inverse of the sum of the values of the function of comparison of the « m » chosen samples.

According to the quality of the solutions, the bees decide either to be loyal to their respective solutions or to abandon it, this choice is taken by the use of the RWS on the probability of whether a bee is loyal to it solution.

The bees split into two groups: the free bees and the recruiters, then we apply the RWS to the probabilities that a recruiter can be chosen, each freebee follows a recruiter and thus it will have the same partial solution, and the first step is done.

In the next step, each bee chooses the next « m » point alone in order to form another partial solution. After the second step, the partial solution will have « 2*m » components and the algorithm continues « N » time until the number of components reach « p ». $N \times m = p$ (2)

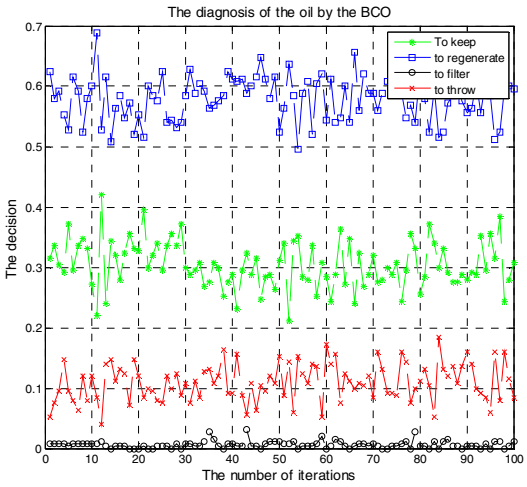


Fig. 4. The diagnosis of the sample no 2 by the BCO

Each bee will have a complete solution; but according to the algorithm, all the bees converge to the same solution.

In this algorithm, there is an evolution of the partial solution with the number of steps, which is not necessary to display since we only need the complete solutions.

But considering the randomness of the RWS, there will be a variation of the percentage of the solution from a computation to another, so we run the algorithm several times and we observe the rate of variation.

We apply the BCO on the sample 2 of the Table 1, and the results are represented in the Fig. 4.

The results of the diagnosis of the four samples are shown in Table 2.

Table 2. Diagnosis of the four Samples by the BCO

	keep	regenerate	filter	discard
E1	62.14%	10.61%	01.33%	25.92%
E2	31.21%	56.90%	00.57%	11.32
E3	15.32%	15.33%	39.93%	29.42%
E4	09.86%	17.58%	32.49%	40.07%

We applied the BCO to the same samples as in the ACO; we notice that the results agree with the decision of the experts.

Comparing the results, the diagnosis by the ACO seemed more precise, but it doesn't show the same complex behavior as the BCO, although it uses the accumulation of pheromone during several iterations. The advantage of BCO is that it reaches the solution in a one iteration using a relatively complex communication behavior.

The BCO uses different strategies of search in reference to the ACO, we can say that it performs a search in space of partial solutions but it is less precise than the ACO.

4 The Hybrid Algorithm ACO/BCO

We have developed two algorithms earlier, the ACO and the BCO, and we tried to use the same reformulation of the problem so that we can compare the results and even to hybrid them. [12]

We note that the two algorithms use a different strategy of search, and the two methods provide results to discuss.

The hybrid algorithm benefits from the two algorithms and the artificial hybrid agent has the features of the artificial ants and the artificial bees.

At each iteration, the hybrid algorithm performs a search in the space of partial solutions and then in the set off complete solutions. Using this strategy it will eliminate the problem of the ACO which is the lack in formations during the formation of the complete solution, and we improve the search of the BCO by accumulating the pheromone during several iterations.

The results of sample 2 of the Table 1, is shown in Fig. 5.

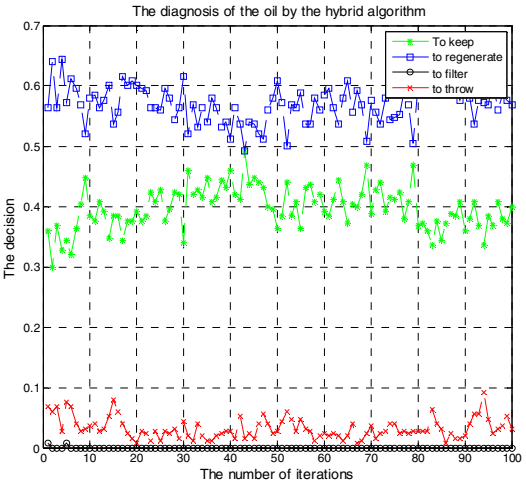


Fig. 5. The diagnosis of the sample 2 by the hybrid algorithm

The hybrid algorithm gives results in agreement with those of the experts. Nevertheless, an objective comparison requires the computation of the exact solution to determine the incertitude.

According to our formulation of the problem, the exact solution is the set of samples from the data base that have the highest value of the function of comparison, or, in other words, the samples that are the most similar to the tested one. We can find it by enumerating all combinations without repetition from the set of samples of the data base, and the exact solution is the one that have the highest value of the function of comparison. . [5]

5 The Adjustment of Parameters of the Algorithms

The exact solution is the set of samples from the data base that have the highest value of the function of comparison, the computation of this later takes excessive time compared to our algorithms.

We adjust the parameters of the algorithms so that the precision is the highest, but this requires another study; nevertheless we did run several tests, in which we change one parameter and fix the others, then take the value that gives the higher precision. The results are shown in Table 3.

Table 3. The implementation parameters of the algorithms

The number of ants	The number of bees	The number of hybrid agents	The evaporation factor	The number of iterations	The parameter α	The parameter β
50	50	50	0.5	20	0.85	0.75

We calculate the error of three algorithms towards the exact solution and the time of computation, for the same sample, the results are shown in this Table 4.

Table 4. Comparison between the Three Algorithms

Algorithm	error	The time of computation (s)
ACO	0.0840	11.0776
BCO	0.2300	0.5689

We notice that the ACO is more precise than the BCO, the time of computation of the BCO is less than the ACO and the hybrid algorithm is relatively fast and the most precise algorithm.

The ACO is more precise than the BCO because the ants find the solution by accumulating the pheromone in several iterations; contrary to the bees that find the solution in one iteration using a complex communication behavior. This gain in precision has it repercussion on the time of computation, because in ACO, the pheromone cannot be updated until all ants finish their tours.

The hybrid algorithm combines the two strategies of search, so the evaluation is occurring during and after the formation of the tour, which allows to find more precise solutions in relatively smaller time of computation.

6 Conclusion

The reliable diagnosis of transformer's oil is primordial to ensure the continuity of the service. In this chapter we have elaborated an adaptation of the ant colony optimization and the bee colony optimization in the diagnosis of the transformer's oil. The elaborated algorithm depends on several parameters, and a judicious choice assures a height precision and a small time of computation.

The hybrid algorithm has better convergence than the ACO and the BCO.

The difficulty of these algorithms lies in the initialization.

References

1. Rolland, N., Magnier, P.: Explosions et Incendies de Transformateurs. Méthode d'Evaluation des Couts, Paris, France, pp. 7–15 (2002)
2. Perrier, C.: Étude des huiles et des mélanges à base d'huile minérale pour transformateurs de puissance – recherche d'un mélange optimal. Thèse de Doctorat, École Centrale de Lyon, France, pp. 1–5 (Avril 2005)
3. Aouchar, N., Bekhaled, C.: Application des systèmes hybrides neuro-flous au diagnosis des huiles de transformateur. Projet de Fin d'Etudes, Département Génie électrique, Ecole Nationale Polytechnique, Alger, pp. 45–50 (2005)
4. Bonabeau, E., Théraulaz, G.: L'intelligence en essaim. Pour La Science (271), 66–70 (2000)

5. Wong, L.P., Low, M.Y., Chong, C.S.: Bee Colony Optimization with Local Search for Traveling Salesman Problem, pp. 1–7. Singapore Institute of Manufacturing Technology, Singapore (2010)
6. Monmarché, N.: Algorithme de fourmis artificielles-application à la classification et à l'optimisation. Thèse de Doctorat, Université François Rabelais Tours, France, pp. 5–45 (Décembre 2000)
7. Tfaili, W.: Conception d'un algorithme de colonie de fourmis pour l'optimisation continue dynamique. Thèse de Doctorat, Université Paris 12, Val de Marne, pp. 6–11 (2007)
8. Bonabeau, E., Dorigo, M., Théraulaz, G.: Swarm intelligence, pp. 1–8. Oxford University Press (1999)
9. Yu, X., Gen, M.: Introduction to Evolutionary Algorithms, pp. 11–31. Springer (2010)
10. Teodorovic, D., Selmic, M.: Bee Colony Optimization:The Applications Survey. Faculty of Transport and Traffic Engineering, University of Belgrade, pp. 1–8 (2011)
11. Farooq, M.: Bee-Inspired Protocol Engineering From Nature to Networks, pp. 147–184. Springer, Heidelberg (2009)
12. Shuang, B., Chen, J., Li, Z.: PS-ACO Hybrid Algorithm: Study on hybrid PS-ACO algorithm. Springer Science+Business Media

Performance Aspect of SaaS Application Based on Tenant-Based Allocation Model in a Public Cloud

Wojciech Stolarz and Marek Woda

Instytut Informatyki, Automatyki i Robotyki, Politechnika Wrocławska
voytec0dh@gmail.com, marek.woda@pwr.wroc.pl

Abstract. The main aim of current study was to check performance of tenant-based vs. tenant-unaware resource allocation model of SaaS application in a public Cloud. In order to do so, two SaaS systems were developed. First of them used traditional resource scaling based on number of users. It acted as a reference point for the second system. Previously conducted tests were primarily focused on measuring over- and underutilization in order to compare cost-effectiveness of the solutions. The tenant-based resource allocation model proved to decrease system's running costs. It also reduced the system's resources underutilization. Similar research was done, but the model was tested in a private cloud. In this paper the systems were deployed into a commercial public cloud and performance aspects were scrutinized.

1 Introduction

Software-as-a-Service is a software distribution paradigm in cloud computing and represents the highest, software layer in the cloud stack. Since most cloud services providers charge for the resource use it is important to create resource efficient applications. One of the way to achieve that is multi-tenant architecture of SaaS applications. It allows the application for efficient self-managing of the resources. In this paper the influence of tenant-based resource allocation model on performance of SaaS systems is investigated. The tenant-based resource allocation model is one of the methods to tackle under-optimal resource utilization. When compared to traditional resource scaling it can reduce the costs of running SaaS systems in cloud environments. The more tenant-oriented the SaaS systems are the more benefits that model can provide.

One of recent solutions for over- and underutilization problems may be a *tenant-based resource allocation model* (*TBRAM*) for SaaS applications. That solution was introduced and tested with regard to CPU and memory utilization in a private [6] and a public cloud [10, 11, 12]. They proved the validity of *TBRAM* by reduction of used server-hours as well as improving the resources utilization. The main aim of the paper is further *TBRAM* approach validation (in terms of sheer technical performance), as a continuation of research part from [10, 11, 12].

Despite that in the cloud one can automatically receive on-demand resources one can still encounter problems related to inappropriate resource pool at the time. These are over- an underutilization [10, 11] which exists because of not fully elastic pay-per-use model used nowadays [9]. Over provisioning exhibits when, after receiving additional resources (in reply for peak loads), they are being kept even if they are no

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_41,

longer needed. Thus we are affected from underutilization. Under provisioning (saturation) exhibits when we cannot deliver required level of service because of insufficient performance. This is also known as an overutilization. It leads to the customers' turnover and revenue losses [2]. For example *Amazon Elastic Cloud Computing (EC2)* service charge users for every partial hour they reserve each *EC2* node. Paying for server-hours is common among cloud providers. That is why it is very important to utilize fully given resources in order to really pay just for what we use.

2 Related Work

Authors in [4] propose profiles approach to scaling in the cloud. They try to use best practices and their knowledge in order to create scalable profiles. The profile contains information that helps to characterize a server in terms of its capabilities. When the scaling activity is fired it takes the profile information into account. In [7] authors propose a toolkit using Java mechanism to support multi-tenancy. They use context elements to track applications running on Java Virtual Machine. That in turn allows distinguishing each tenant. That information can be later used in order to estimate given tenant's resource usage. The tenant context can also be used for billing each tenant's activities. In [5] authors consider an intelligent resource mapping as well as an efficient *virtual machines* (*VM*) management. It is a very important problem that greatly influences costs of running applications in a cloud. In [8] authors describe three major components which influence virtual machines performance. These are: measurement, modeling and resource management. They introduce a decomposition model for estimating potential performance loss while consolidating *VMs*. Amazon proposes its *Auto Scaling* tool [1] to manage *VM* instances using predefined or user-defined triggers. It is the *Amazon EC2* platform specific mechanism based on resource utilization. In [6] authors implements a tenant-based resource allocation model for their SaaS application deployed in private Eucalyptus cloud. The authors performed tests with incremental and peak workload simulation. In the research they achieved significant reduce of server-hours compared to traditional resource scaling model. The tenant-based model improved also utilization of cloud resources by their SaaS system. Moreover, they introduce formal measures for under and over provisioning of virtual resources. The measures are designed specifically for SaaS applications with respect to CPU and memory utilization. In this paper cost-effective tenant-based resource allocation model of SaaS system is presented. Their publication will be referred to as the base paper in the remaining parts of this paper.

3 Systems

3.1 Base System

Through the publications [6, 10, 11] the base tenant-unaware resource allocation SaaS system (Base System) was described. It conforms to a traditional approach to scaling resources in a cloud and is based on number of users of the system. Load-balancing technique leverage a simple but fast and even round-robin algorithm offered by *Apache HTTP*. It is substituted by Amazon *Elastic Load Balancer* service. According

to [3] the *Elastic Load Balancer (ELB)* sends special requests to balancing domain's instances to check their statuses (health check). Then it round-robins among healthy instances with the less number of outstanding requests. In the process it does not take into consideration the instances resource usage of any kind. Although the name of this system suggest lack of awareness of tenants it concerns only resource allocation.

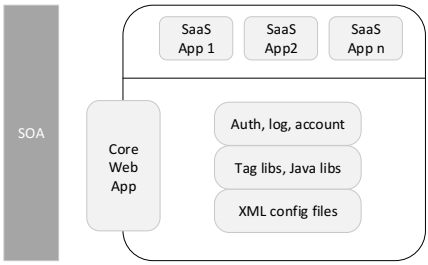


Fig. 1. SaaS platform architecture

The system was build according to *Service Oriented Architecture (SOA)* and native multi-tenancy pattern. First, it was implemented as a set of J2EE web applications using *Spring* and *Struts* frameworks. Several parts of the system were later transformed to web services using *WSO2* and *Axis2*. Deploying application as a web service makes it independent from running platform. It also gives more flexibility with accessing the application.

3.2 Tenant Aware System

The base SaaS system was implemented as a reference tenant-unaware resource allocation system. The main flaw of its design was rigid management of VM instances in the cloud. Thus, it could lead to serious over- and underutilization problems, that judgment will find justification in the chapter with test results. One of the ways to tackle these problems was proposed [6] a tenant-based resource allocation model (TBRAM) for scaling SaaS applications over cloud infrastructures. By minimizing utilization problems it should decrease also the final cost of running the system in the cloud. The *TBRAM* consists of three approaches that leverage multi-tenancy to achieve its goals. The first of them is tenant- based isolation [10, 12], which separates contexts for different tenants.

It was implemented with tenant-based authentication and data persistence as a part of the SaaS platform (Tomcat instances). The second way is to use tenant-based VM allocation [10, 12]. With that approach authors was able to calculate the actual number of needed VMs by each tenant in given moment. The last but not least is the *tenant-based load balancer* [11] that allows to distribute virtual machines' load with respect to certain tenant. An overview of the architecture is presented in Fig. 1 below. We can notice that the *SaaS Core Web App (SCWA)* element in the Fig. 2 was the only change made to the original test bed [6]. That authorial element embraced proposed *TBRAM* approach [10].

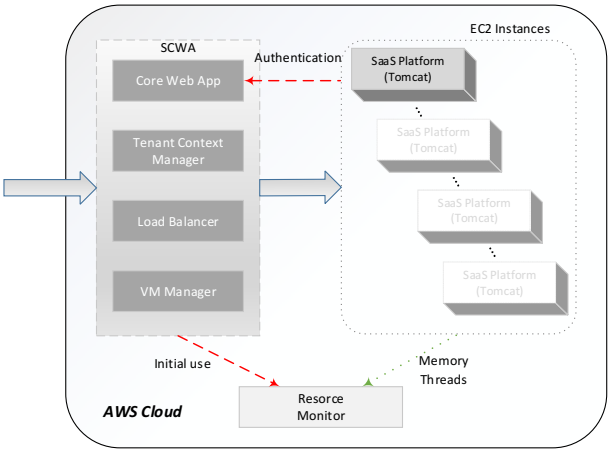


Fig. 2. TBRAM system architecture

4 Performance Analysis

Tests to compare authorial SaaS systems built in accordance to *TBRAM* (and without it) were designed and conducted. The results in terms of server-hours, over- / underutilization and financial cost were presented already in [10, 11, 12].

These results gave us the systems' behaviour comparison according to the base paper [6] methodology. In this part of the paper we describe in more detail the systems performance recorded during the tests. To visualize the both systems behaviours graphical charts from the *Amazon CloudWatch* web application were used. Let us first take a look at the entry points to the systems which were the load balancers. In the charts (Fig. 3, 4 ,5) we can see the Base System components performance during the incremental workload test. The components are showed in sequence they were traversed by simulated client requests. That is from the *ELB* to the group of *SaaS* platform instances and eventually to the database. In the Fig. 3 we can notice the moment when the *ELB* was scaled up. At about 08:00 August 26th the number of incoming requests to the ELB was still rising while the latency dramatically decreased because of the scaling activity.

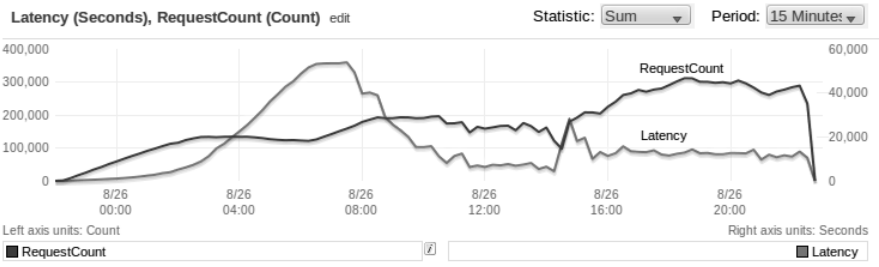


Fig. 3. ELB performance during Base System incremental test

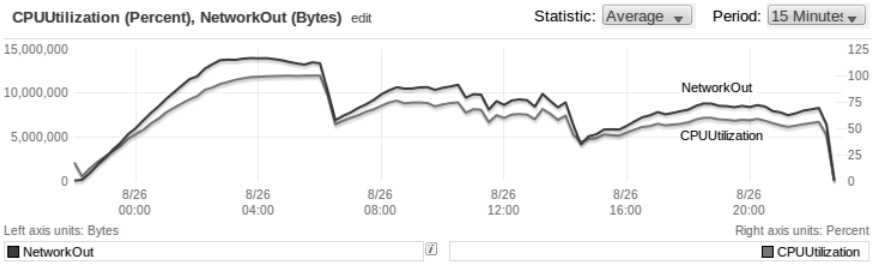


Fig. 4. SaaS platform instances group performance during Base System incremental test

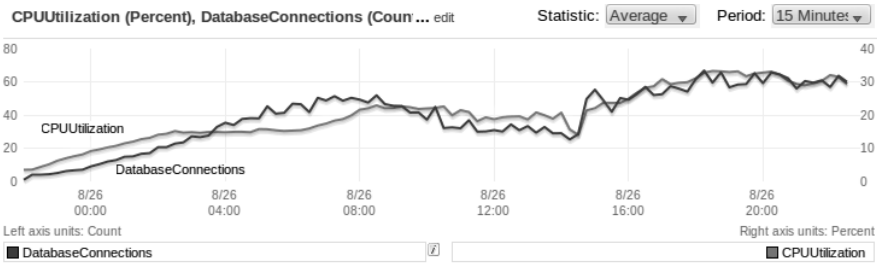


Fig. 5. DB performance during Base System incremental test

By looking at the sum of requests number we can distinguish three periods during which the *VMs* number was different (2, 4 and 8 *VMs* in this case). When the *VMs* number was increased we can notice rapid increase of requests handled by the *ELB* that appeared about 7am and 3pm on the chart. Fig. 4 presents an aggregated performance of the *SaaS* platform instances group. We can also notice 3 distinguishable parts of the test. Each time new instances were started we can observe decreased overall utilization by the instances. This is because newly started *VMs* handle some part of the workload therefore decreasing the overall group utilization. The third chart (Fig. 5) is just to show that the database (*DB*) was not a bottleneck during the test. We can see that the *CPU* utilization by the *DB* rarely exceeded 30%, even under the full system load at the end of the incremental test. Similarly the number of connections to the *DB* rarely exceeded 60 out of max. 125. Due to the strict connection release policy of the persistence layer.

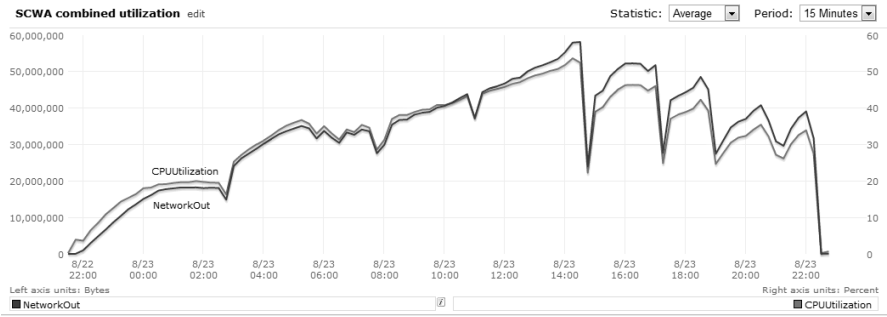


Fig. 6. SCWA performance during TBRAM system incremental test

Let us take a look now at the *TBRAM* system's performance during the incremental workload test. The following charts (Fig. 6, 7, 8) presents the system's behaviour. The main difference between the charts is visible in the first chart (Fig. 6). Instead of latency and request count statistics typical to *ELB* we see standard *EC2* instance metrics. We can observe strong relation between the *CPU* utilization and the throughput. On the second chart (Fig. 7) we can see that periodical drops of the SaaS instances group resources utilization. They represent the moments where new instances were added to existing group. We do not see 3 distinguishable parts any more since the *VMs* were added dynamically when needed.

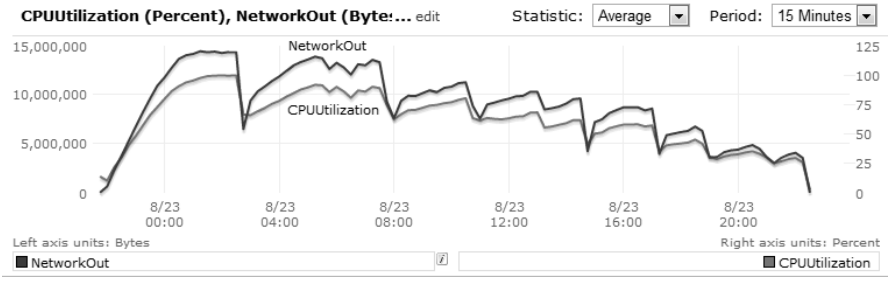


Fig. 7. SaaS platform instances group performance during TBRAM system incremental test

The last chart (Fig. 8) was presented just to be consistent with the Base System charts. We can see that the maximal number of concurrent connections increased to about 85 while the maximal *CPU* utilization remained at the same level. Once again we do not see any symptoms of the database saturation. Therefore, we can assume that the *DB* was not a bottleneck for the system and did not negatively influenced the results. We based our assumption also on other *DB* metrics like read/write latency, *IOPS* and throughput. Now it is time for the systems performance comparison during the peak-based workload test. The following two charts (Fig. 9, 10) visualize the Base System behaviour. Once again we can see three distinguishable parts of the test, only this time the parts duration period varies. The reason why is the test's constraint for the number of iterations rather than for the test duration. That was explained before in the test description part of this paper.

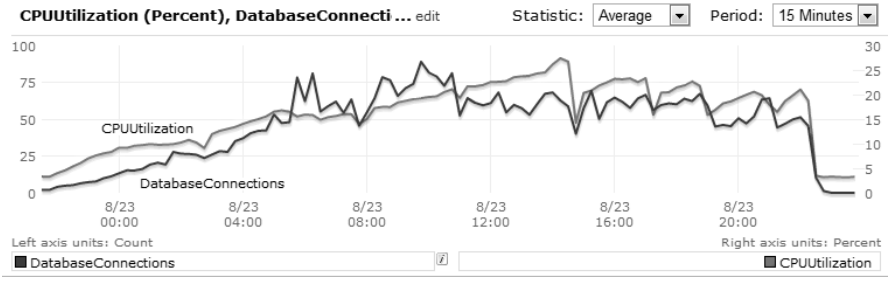


Fig. 8. DB performance during the TBRAM system incremental test

By looking at the Fig. 9 we can notice the relation between the peek number of simulated users (in the middle of each the test's part) and the *ELB* latency. The second chart (Fig. 10) shows the SaaS platform instances group behaviour. We can observe the relation between the *ELB's* request count and the group's resources utilization.

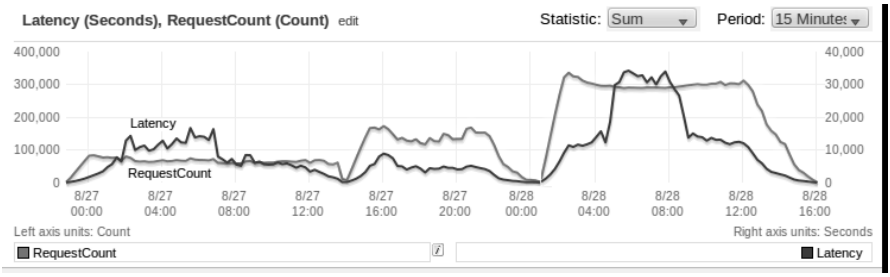


Fig. 9. ELB performance during the Base System peak-based test

Finally, let us take a look at the TBRAM system. The following two charts visualize the system performance similarly to the Base System (Fig. 10, 11) In the first chart (Fig. 10) we can clearly see the SCWA element utilization reflected peak-based workload. We can observe 3 distinct parts of the test. The last part is little longer than the first two. That is even despite this test's characteristic were set up to make each of the parts last the same time period. As mentioned before, the peak-based tests were constrained with the number of iterations. Because during the third part of the test the system did not behaved as it was expected (plateau instead of peak) the test duration was stretched out. We can see the plateau also on the second chart (Fig. 12) during the third test's part. We can observe that the SaaS group was not saturated by the look at the metrics, so the group's performance was not the reason of the plateau.

The problem arose with finding the source of the plateau. Our first guess was the database saturation. After all, the number of concurrent users increased very rapidly to 400 during the third part of the test. It was possible though to exceed the maximal number of *DB's* concurrent connections.

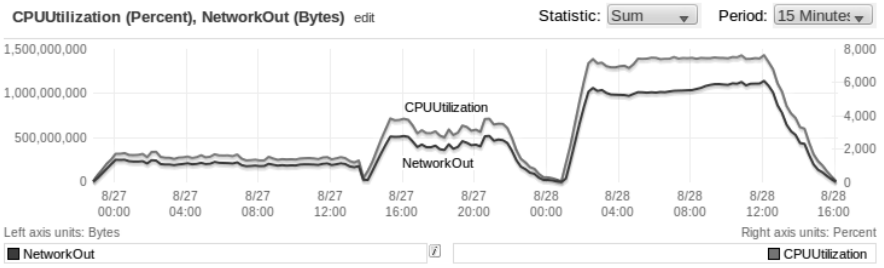


Fig. 10. SaaS platform instances group performance during the Base System peak-based test

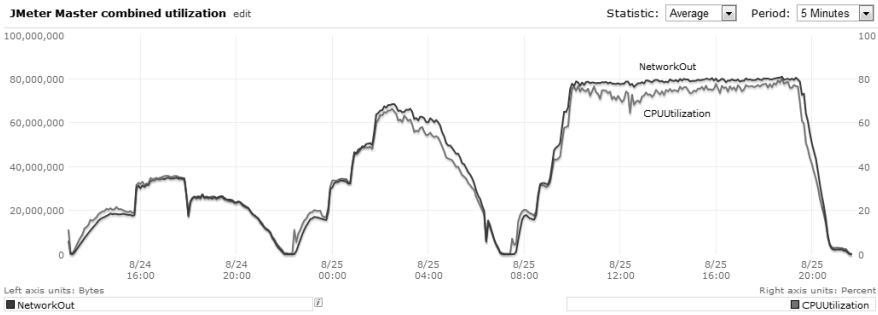


Fig. 11. SCWA performance during the TBRAM system peak-based test

Checking the *DB's* metrics however did not confirmed that theory. So the only part of the system that left was the *JMeter* cluster used to generate all the workload. The corresponding charts is shown below:

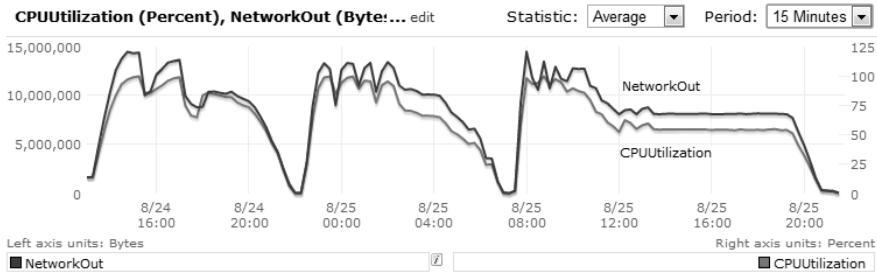


Fig. 12. SaaS platform instances group performance during the TBRAM system peak-based

Fig. 13 shows the average aggregated resource utilization by the *JMeter* auto-scaling group of instances. We need to keep in mind that the metrics where aggregated. It means that even the average metrics for the first two parts of the test looks similar, the throughput of the group was actually different. It depended mainly on the *JMeter* slave instances number in given test part, which was 1, 2 and 4 respectively. More to the point, we can clearly see that the utilization during the last test's part looks different. It does not, however, exhibits any saturation symptoms. Finally, let us examine the *JMeter* master instance performance.

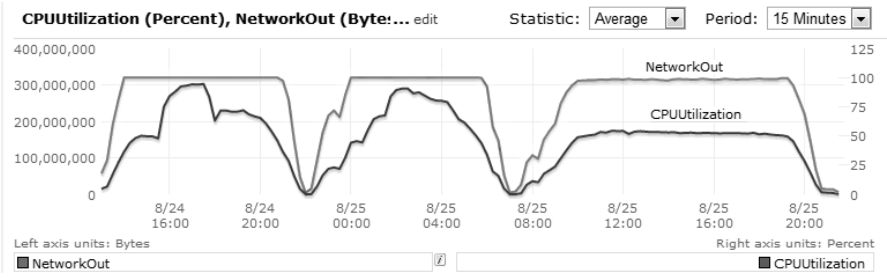


Fig. 13. JMeter slave instances group performance during the TBRAM peak-based test

In Fig. 14 we can see the resource usage by the *JMeter* cluster's master instance. On the chart we can see that during the second part of the test the *JMeter* instance was already close to saturation. During the last part it exhibits saturation symptoms because of the CPU overutilization. This kind of instance behaviour was not expected by me. The good part is that the very same instance was used in all other tests. Thus, the tests were conducted in the same conditions making the results valid. Note: Please ignore the network out peak at the beginning of the third part of the test. It was caused by the *JMeter's* test data upload to the *Amazon Simple Storage Service* (S3) server. It took just a minute and did not affected the test.

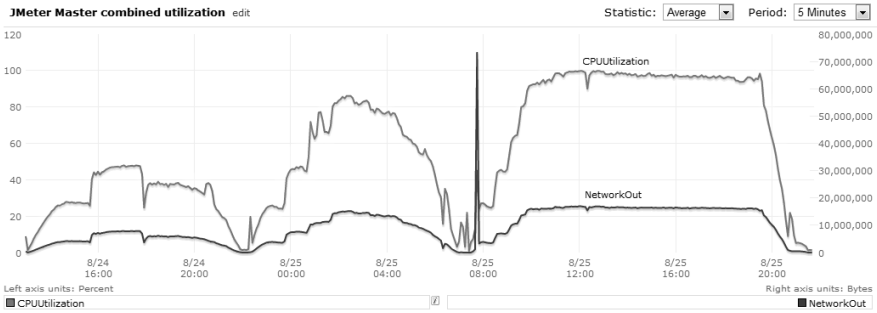


Fig. 14. JMeter master instance performance during TBRAM system peak-based test

5 Conclusions

The *TBRAM* system introduced a twofold improvement: tenant-based load-balancing and tenant-based resource scaling. Dynamic resource scaling based on current tenants needs, can significantly reduce server-hours used by multi-tenant SaaS cloud systems. It achieves that by avoiding the resource over-provisioning typical to the traditional resource scaling (based on current users number). Therefore, it is being thought that most *SaaS* systems would benefit from the *TBRAM* in that matter, since it does not depend on the system's type. However, possible improvement of tenant-based load-balancing compared to traditional load-balancing (for example based on round-robin algorithm) depends more on the type of the SaaS system. If the *SaaS* system's applications are using significant amount of tenant related data, then dispatching each tenants workload to the same VMs can take advantage of server's caching. That can increase the server's performance. On the other hand, if the SaaS applications does not use tenant related data so extensively, then simple, fast and even load-balancing could be sufficient. Of course the results also depend on the *TBRAM* implementation.

This work was inspired and based on the base paper [6], in which the authors achieved 32% server-hours reduction compared to traditional resource scaling. In current research authors achieved about 20% and 30% reduction in case of incremental and peak-based tests respectively. The better result for the peak-based test is caused mainly by the *TBRAM* underutilization improvement achieved for that type of workload. In the base work the model statistically improved also just the

underutilization, but for both types of workload. Thus, this work confirms the *TBRAM* benefits making it worth to consider even more.

References

1. Amazon Auto Scaling, http://aws.amazon.com/autoscaling/ (accessed: January 27, 2014)
2. Armbrust, M., et al.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report #UCB/EECS-2009-28. Electrical Engineering and Computer Sciences University of California at Berkeley (2009)
3. AWS Developer Forums: ELB Strategy: https://forums.aws.amazon.com/thread.jspa?messageID=135549 (accessed: January 27, 2014)
4. Guo, C.J., et al.: A framework for native multi-tenancy application development and management. In: 2007 9th IEEE International Conference on e-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, e-Commerce, and e-Services, Piscataway, NJ, USA, July 23-26, pp. 470–477 (2007)
5. Chen, Y., et al.: An Efficient Resource Management System for Online Virtual Cluster Provision. In: IEEE International Conference on Cloud Computing, CLOUD 2009, pp. 72–79 (September 2009)
6. Espadas, J., et al.: A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures (2011)
7. Cai, H., et al.: An end-to-end methodology and toolkit for fine granularity SaaSization. In: 2009 IEEE International Conference on Cloud Computing (CLOUD), Piscataway, NJ, USA, September 21-25, pp. 101–108 (2009)
8. Iyer, R., et al.: VM3: Measuring, modeling and managing VM shared resources. Computer Networks 53(17), 2873–2887 (2009)
9. Stillwell, M., et al.: Resource allocation algorithms for virtualized service hosting platforms. Journal of Parallel and Distributed Computing 70(9), 962–974 (2010)
10. Stolarz, W., Woda, M.: Proposal of cost-effective tenant-based resource allocation model for a SaaS system. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 409–420. Springer, Heidelberg (2013)
11. Stolarz, W., Woda, M.: Tenant-Based Resource Allocation Model application in a public Cloud. IEEE Transactions on Industrial Informatics (2014) (submitted)
12. Stolarz, W., Woda, M.: Zwiększenie efektywności aplikacji typu Saas poprzez wykorzystanie modelu alokacji zasobów bazowanego na dzierżawie zasobów najemcom. Studia Informatica 34(3), 223–238 (2013)

Low Cost FPGA Devices in High Speed Implementations of KECCAK-*f* Hash Algorithm

Jarosław Sugier

Wrocław University of Technology, Institute of Computer Engineering, Control and Robotics
jaroslaw.sugier@pwr.wroc.pl

Abstract. Cryptographic hash functions are important components in many applications of contemporary information systems like computation of digital signatures, authentication codes or fingerprinting. The recent SHA-3 competition announced by NIST resulted in developments of new hash methods from which the Keccak algorithm has been selected as the winner after intensive public evaluation of the candidates. In this paper we are discussing various high-speed organizations of the Keccak-*f*[1600] permutation function – the core component of the Keccak SHA-3 algorithm – which can be created from the basic iterative architecture by round replication (loop unrolling) and pipelining. Different variants of the proposed architectures are implemented in a popular Spartan-3 device form Xilinx and the presented results identify main problems which arise if a high speed architecture of the function is automatically implemented in an FPGA device.

Keywords: configurable hardware, SHA algorithm, pipelining, iterative architecture.

1 Introduction

The SHA-3 competition, announced by the U.S. National Institute of Standard and Technology (NIST) in November 2007 ([10]), ignited a lot of new research on hash functions. From the 5 selected finalists, the Keccak algorithm proposed by Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche was eventually selected as the winner and at the time of this writing is about to be announced as the official new SHA-3 standard.

In this work we will discuss hardware implementations of the Keccak-*f*[1600] permutation function – the essential component of the algorithm used for hash calculation – in high speed organizations built around the concepts of loop unrolling and pipelining. The 6 proposed variants of the processing will be implemented in a popular Spartan-3 device from Xilinx, creating a consistent base for evaluation of both the hardware organization concepts and efficiency of automatic implementation on the FPGA platform.

Organization of the text is as follows. In the next chapter we will relate to extensive research on Keccak hardware implementations done during the SHA-3 competition and present motivation for the work in this paper. Then, in chapter 3, we will

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_42,

introduce loop unrolled and pipelined architectures being under evaluation here and in chapter 4 we will discuss the results obtained after their implementation in the FPGA chip.

2 Related Work

2.1 Realization of Keccak in Hardware

Efficiency in both software as well as hardware implementation was, next to cryptographic strength, the key aspect taken into account by the Keccak's authors during development of the method [5]. Regular, round-based structure of the processing and simplicity of the elementary transformations (which purposely do not include any substitution boxes neither multi-bit additions, unlike most of the state-of-the-art block ciphers of the AES class) were intentionally adopted in order to facilitate implementations in hardware. This was also in line with requirements put forward by NIST already in the initial SHA-3 competition call [10] which among requirements for candidate submissions listed "a statement of the algorithm's estimated computational efficiency and memory requirements in hardware". During extensive public examination of all SHA-3 candidates a lot of efforts were devoted to this aspect and there is abundant amount of information about efficiency of various Keccak[1600] architectures in both mask-programmable (ASIC) and user-programmable (FPGA) digital electronic devices.

The official author's statement ([2]) about Keccak hardware implementations describes two reference VHDL designs with different speed vs. area trade-offs: the "high speed" stand-alone core and the "mid-range" coprocessor. The high speed core was built upon implementation of one complete round of the hash in hardware where the block of state bits was iterated in a series of clock cycles equal to the number of rounds, i.e. 24. This kind of organization will be denoted as x1 in this work. The low-area coprocessor made use of external (system) memory for storage of the state bits and was suitable for embedded environments like smart cards or wireless sensor networks where area and power savings are particularly important. In [2] both of the designs were implemented as ASIC devices using STMicroelectronics 130 nm technology while results of their FPGA implementations were given in [11].

2.2 Different Architectural Options

The most comprehensive database of various FPGA implementations of Keccak (and other cryptographic algorithms, including all other SHA-3 candidates) has been built around ATHENa project at George Mason University available at [1]. An "Automated Tool for Hardware EvaluatioN" was developed to create an open-source environment for fair, comprehensive, automated, and collaborative hardware benchmarking of algorithms belonging to the same class ([8]). It was the platform used by a group of researchers in comprehensive evaluation of all SHA-3 contenders with regard to their

FPGA effectiveness. The conclusions of their studies were published in [7] and were thoroughly discussed during the phase of public evaluation within the SHA-3 contest.

In taxonomy of that work, taking as the starting point the plain iterative organization (one round implemented in hardware and transformation of the state data consisted in a loop of n_r iterations) the two opposing techniques can be used to create various derivate architectures with different area vs. speed trade-offs: loop unrolling and round folding. In loop unrolling more than one round is instantiated in hardware so the number of loop iterations is reduced and thus the speed of data processing is increased, while in round folding either only part of the round in included in the hardware block (so called horizontal folding) or only part of the state is processed in the block (so called vertical folding). In both variants of folding the computation of one round takes multiple clock cycles (slower processing as a cost of reduced area) while in loop unrolling the extra hardware returns in increased throughput. Additionally, each of these techniques can be enhanced with pipelining if there is a cascade of functional modules and multiple data blocks can be processed at the same time. Meeting the latter condition depends on operational environment and is possible only if multiple messages from the input stream can be hashed simultaneously.

Apart from these generic variations of the basic iterative architecture another intrinsic optimizations of the Keccak core processing steps can be proposed which would bring additional benefits especially in cases of low-throughput area-sensitive designs. For example, in [9] an original re-arrangement of round operation was proposed with the intention to implement vertical round folding by a factor of 8: the entire 24-round processing was re-partitioned into new 25 rounds so that the order of elementary transformations within each one could be modified. Then, with the 1600b of state stored in 25 8x8 distributed RAM modules and 1/8 of the state processed in each clock cycle, all the rounds could be computed in 200 clock cycles – with substantial savings in design size.

2.3 Scope of This Work

In this paper we will evaluate effectiveness of automatic implementation of the high-speed Keccak-*f*[1600] architectures built around the concepts of loop unrolling and pipelining in popular FPGA devices from Xilinx. We will present results obtained after implementation of the basic iterative architecture (x1) and compare it to the effects of loop-unrolled organizations with two (x2), three (x3) and four (x4) rounds implemented in hardware, with and without pipelining after each round. The unrolled pipelined organizations will be denoted as x*k*-PPL*k*, after [7].

The term “low-cost” that we refer to in the title and in this text is understood not only as using inexpensive devices as the target hardware, but also as trying to minimize the cost of the design and its automatic implementation. In contemporary systems the cryptographic unit often becomes just one of the elements of the entire system and it is not desirable, or even not possible, to make its optimization to be the dominant aspect of the whole FPGA project. The module must share both the resources and the optimization effort appropriately with the rest of the system. In such a situation not only the performance of the unit (generally understood almost always as

maximum data throughput per occupied area) but its flexibility and fast, fully automatic implementation become highly valued features that facilitates installation of the cipher in the whole design and, consequently, reduce time-to-market in device development. On such ground, from the perspective of this work, it is interesting to evaluate effectiveness of the software tools in automatic implementation of the Keccak algorithm in the above mentioned architectures.

3 High-Speed Implementations of the Keccak-*f* Function

3.1 Specification of Keccak-*f*

Keccak-*f*[*b*] is a family of seven permutation functions: for $l = 0, 1, \ldots 6$ each function operates on a state of $b = 25 \times 2^l$ bits (b = 25, 50, 100, 200, 400 800, and 1600) where a single word of $w = 2^l$ bits is called *a lane*. Every function computes its result processing the state in a series of n_r rounds, $n_r = 12 + 2l$ (n_r = 12, 14, 16, 18, 20, 22, and 24). The rounds are identical but they apply different w-bit constants in their final transformation. For the SHA-3 contest the strongest (and the largest) version of Keccak[1600] was proposed where 1600 bits of state consisted in 25 64b lanes is transformed in 24 rounds and this version is the subject of this work.

With the use of the selected Keccak-*f* permutation, the final Keccak[*c*, *r*] hash function ($c + r = b$) is built on the fundamental concept of a *sponge construction* ([3]) which, with specific padding, can generate a hash digest of any size for an input stream of arbitrary length. Parameters c (capacity) and r (bitrate) can be adjusted to find the desired balance between speed vs. cryptographic strength of the generated hash.

The reference in [5] describes one round of Keccak-*f*[b] as a sequence of operations on state A which is represented as a 3-dimensional array $A[5][5][w]$. The sequence consists of 5 transformations:

$$\mathrm{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

and each one is defined on individual bits of the state. Computing the permutation is equivalent just to applying the round function n_r times to the input vector each time using a unique w-bit constant RC[i] in the last transformation ι.

3.2 HDL Coding Style

In the reference VHDL specification ([4]) transformations of the Keccak round are expressed as manipulations of the individual bits of the state. While such a description on the lowest level of detail does not enforce any constraints on interpretations made by the implementation tools and may help in efficient synthesis, the code itself is quite long and cumbersome in maintenance.

For the needs of this work an original code was written and it was based on lane-oriented operations as expressed in the specifications summary at [6]. This specification was ported to VHDL language using strict RTL style: there was no instances of

any Xilinx library primitives, no sequential (procedural) descriptions were inserted and no references to any hardware attributes were made so that the code was ready to be synthesized for a different device from any manufacturer. Because the elementary operations include vector rotations in θ and ρ steps, it was decided to use VHDL **`rol`** operator from the standard `NUMERIC_STD` library and because of this all the internal vector signals were defined as `unsigned` type. The only other elementary operations are `and`, `xor` and `not` and these are built-in operators of the language. As a result, the code became much more compact and easier to interpret. The entire round was described in 40 lines instead of 190.

Simplified and more concise description did not cause any significant changes in efficiency of the resultant hardware. As a test, the round module coded with the two styles was used in the basic iterative architecture (x1) implemented in the two families of FPGA devices – Spartan-3 and Spartan-6. The results turned out to be almost identical: the highest difference reached the level of 1% in implementation size expressed in number of occupied slices in the case of Spartan-3 device, but in the same design there was no difference in number of used LUTs so just packing of the LUTs in the slices was a little less effective. The throughputs of both implementations were also very close: the maximum operating frequency of the proposed coding was only by 0.23% higher in Spartan-6 and by 0.55% lower in Spartan-3.

3.3 Implementing Different Computation Schemes

Regular structure of the Keccak-*f* processing with a series of 24 identical rounds makes implementation of the basic iterative architecture x1 straightforward. The hardware needs to include one instance of the round module, simple multiplexing logic at the input (loading either the input data or feeding back the round output) and a counter providing the iteration number. Although in the case of x1 organization it is possible to use an LFSR register for on-the-fly generation of the 64b round constants as it is defined in [5], similarly to [4] the constants were stored in a ROM which was addressed with the round counter. Each of these elements can be expressed in VHDL with a few lines of code. Latency of the complete computation (i.e. the time from loading the input data to reading the output) is 24 clock cycles.

In the case of loop unrolling the only aspect that makes this simple scheme more complex is simultaneous application of multiple round constants. For a x*k* unrolling the latency of computation is 24/*k* and there is a cascade of *k* round blocks in hardware: the first block uses constants RC[0], RC[k], RC[2k]… in the consecutive clock cycles, the second uses RC[1], RC[k + 1], RC[2k + 1]…, etc.. As a result, each round block operates with its own ROM module and uses its own counter for addressing. Because there are 24 rounds in total, the loop can be unrolled by the factors of 2, 3, 4, 6, 8, and 12 – and the first three options from this list are tested in this work.

Pipelining adds more complications to the unrolled architecture. In this paper we investigate cases when each unrolled round creates a separate pipeline stage, i.e. schemes x*k*-PPL*k*, and this returns the latency to the value of n_r (24) clock cycles regardless of the unrolling factor *k*. Creating such organizations needs special attention not because of the presence of the pipeline registers (adding such registers is

trivial) but because of the new rules which need to be applied to addressing of the round constants. Still, the result is based on the configuration where each hardware round operates with its own ROM module and with its own addressing counter – although the counting cycle needs specific redefinition.

4 Implementation Results

All seven architectures: the basic iterative x1 plus 3 unrolled and 3 unrolled and pipelined variants were automatically synthesized and implemented in Xilinx ISE ver. 14.6 software with XST synthesis tool, for a Spartan-3 XC3S2000-5FGG676 device. In all cases the unit computing the Keccak-*f* function was equipped with basic serial input/output shift registers for transferring the 1600b vectors in 64b chunks.

The results are listed in Table 1. Performance parameters were calculated from the minimum clock period estimated by the tools in static timing analysis of the final routed design. Additionally, Table 2 compares essential size and performance parameters of the 6 variants to the parameters of the basic iterative architecture.

4.1 Loop Unrolling

Instantiating multiple rounds in hardware obviously increases resource utilization; scaling of the size in the x2, x3 and x4 architectures can be estimated from the numbers of used LUTs and occupied slices. Actually, the increase is slower than a simple scheme would expected: implementing two rounds takes ×1.57 more slices and 4 – ×2.51. Somewhat higher numbers are recorded for LUTs but they are still not as high as ×2, ×3 and ×4. This only tells that packing the elements in the logic blocks of the FPGA matrix is less intensive for smaller designs, which is usually acceptable.

Unfortunately, not so good conclusions can be drawn from the performance figures. The minimum clock period (or maximum clock frequency) deteriorates in a rate which is much faster than the expected scaling would suggest. Although propagation time of the signal through two rounds (x2 architecture) increases by an acceptable factor of ×2.25, in the x3 architecture the increase is ×4.08 and already ×6.50 in x4 one – so the growth is not only faster than ×*k*, but also nonlinear. As a result the plain unrolling, even with decreased latency in T_{CLK}, leads to lower overall throughput and also to worse throughput per slice values.

This can be explained by looking at the logic vs. routing delay numbers: while the propagation delay generated by logic (LUTs) increases again slower than the scaling would imply, the increase in the routing is by far above the expectations: ×2.47, ×4.87 and ×8.03 and this is the source of unacceptable increase in the minimum clock period. At the same time the average fanout of non-clock nets remains at the equivalent level across all the designs so the nature of their internal combinational functions does not change.

This well illustrates the conclusion that very dense, irregular combinational networks representing the transformations of the large 1600b state over multiple rounds

Table 1. Various architectures implemented in the Spartan-3 device

	Basic iterative x1	Loop unrolled			Loop unrolled & pipelined		
		x2	x3	x4	x2 – PPL2	x3 – PPL3	x4 – PPL4
LUT generators	7017	11868	15071	19399	10816	14607	18472
%	17	29	37	47	26	36	45
Slices	4443	6988	8665	11173	6409	8463	10226
%	22	34	42	55	31	41	50
Registers	4810	4810	4811	4809	6411	8016	9621
Average non-clk. fanout	3.44	3.63	3.42	3.43	3.25	3.24	3.23
Mbps/Slice	1.05	0.59	0.39	0.26	1.28	1.07	0.99
Latency [T_{CLK}]	24	12	8	6	24	24	24
Parallel threads	1	1	1	1	2	3	4
f_{max} [MHz]	102.6	45.6	25.2	15.8	90.8	66.8	55.8
Throughput [Gbps]	4.65	4.13	3.42	2.86	8.23	9.09	10.11
Longest path [ns]:							
logic	2.763	4.729	6.316	8.132	2.663	2.763	2.763
routing	6.861	16.918	33.395	55.098	8.356	12.206	15.174

creates a very challenging task for automatic routing done by the implementation software, even if the overall utilization of the logic in the device remains at the level of 50% or well below. The fact that elementary operations in Keccak include only rudimentary bitwise operators `and`, `xor` and `not` cannot counterweight this fact in the LUT-based FPGA arrays. As a result, plain increase in data throughput of the cipher module cannot be achieved by simple round replication in the hardware like it was in the cases of other, smaller ciphers like AES, Serpent or Salsa20 ([12-14]).

4.2 Effects of Pipelining

Adding pipeline registers is usually well absorbed by the FPGA array and the Keccak case provides a good example: although the number of flip-flops in x*k*-PPL*k* designs are proportionally higher than in corresponding x*k* ones, there is no increase at all in the numbers of used logic blocks (slices) and, actually, additional registers spread uniformly over the networks improve the placement and produce a slight decrease in this parameter.

From the performance point of view pipelining shortens combinational paths radically and, since the state bits are registered on the output of each round, the situation remains comparable to the basic iterative architecture with just one round. Unfortunately

Table 2. Relative parameters of the tested architectures; value for the basic iterative x1 organization = 1.00

	x2	x3	x4	x2 – PPL2	x3 – PPL3	x4 – PPL4
Slices	1.57	1.95	2.51	1.44	1.90	2.30
LUT generators	1.69	2.15	2.76	1.54	2.08	2.63
Minimum T_{CLK}	2.25	4.08	6.50	1.13	1.54	1.84
Longest path: logic	1.71	2.29	2.94	0.96	1.00	1.00
route	2.47	4.87	8.03	1.22	1.78	2.21

the increase of the routing delay with the number of unrolled rounds is still visible, although it is not as strong as in the above case. Nevertheless, this time one could expect that such an increase should not happen: if the rounds are regularly replicated with the pipeline registers on their outputs, the in-round routing (so the longest combinational paths) should have the same complexity as in the basic iterative organization. Because this is not met, the minimum clock period is not constant and the maximum operating frequency decreases. Fortunately, increase in speed coming from the parallel processing of multiple pipelined data prevails significantly and the overall throughput numbers climb, making the pipelined architectures the best ones with regard to this parameter.

5 Conclusions

This work provides comparison of high speed architectures of the Keccak-*f*[1600] permutation function implemented in a low-cost Spartan-3 FPGA device. In order to achieve high data throughputs the two standard methods: loop unrolling and pipelining were applied in two series of architectures. Having three cases in each series it was possible to compare efficiency of the methods as compared to the basic iterative solution and to evaluate how they scale in both size and performance in the hardware.

The results illustrate difficulties which Keccak creates in automatic implementation on the FPGA platform. The trend in development of the FPGA arrays is towards more complex logic blocks – nodes of the array – which are capable of implementation of more and more involved combinational functions, but the progress in the routing capabilities is not as strong. Very dense, irregular combinational networks of Keccak elementary transformations, even built from simple bitwise operations, if mapped to FPGA LUT generators very quickly lead to routing congestion and low performance parameters when the number of implemented rounds increases. In such a situation pipelining – which splits long combinational paths with registers – does offer significant performance improvements with very little or even null cost in size.

References

1. ATHENa Database of FPGA Results, http://cryptography.gmu.edu/athenadb/fpga_hash
2. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: Keccak implementation overview, http://keccak.noekeon.org (retrieved March 2014)
3. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Cryptographic sponge functions. PDF file, http://keccak.noekeon.org (retrieved March 2014)
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Hardware implementation of Keccak in VHDL. KeccakVHDL-3.1.zip, http://keccak.noekeon.org (retrieved March 2014)
5. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak reference. PDF file, http://keccak.noekeon.org (retrieved March 2014)
6. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak sponge function family, http://keccak.noekeon.org (accessed March 2014)
7. Gaj, K., Homsirikamol, E., Rogawski, M., Shahid, R., Sharif, M.U.: Comprehensive evaluation of high-speed and medium-speed implementations of five SHA-3 finalists using Xilinx and Altera FPGAs. In: The Third SHA-3 Candidate Conference. Available: IACR Cryptology ePrint Archive, 2012, 368 (2012)
8. Gaj, K., Kaps, J.P., Amirineni, V., Rogawski, M., Homsirikamol, E., Brewster, B.Y.: ATHENa – Automated Tool for Hardware EvaluatioN: Toward Fair and Comprehensive Benchmarking of Cryptographic Hardware Using FPGAs. In: 20th International Conference on Field Programmable Logic and Applications, Milano, Italy (2010)
9. Junkg, B., Apfelbeck, J.: Area-efficient FPGA implementations of the SHA-3 finalists. In: 2011 International Conference on Reconfigurable Computing and FPGAs (ReConFig), pp. 235–241. IEEE (2011)
10. National Institute of Standards and Technology: Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA–3) Family. U.S. Federal Register vol. 72(212), pp. 62212–62220 (2007)
11. Strömbergson, J.: Implementation of the Keccak hash function in FPGA devices, http://www.strombergson.com/files/Keccak_in_FPGAs.pdf (retrieved March 2014)
12. Sugier, J.: Implementing AES and Serpent ciphers in new generation of low-cost FPGA devices. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Complex Systems and Dependability. AISC, vol. 170, pp. 273–287. Springer, Heidelberg (2012)
13. Sugier, J.: Implementing Salsa20 vs. AES and Serpent Ciphers in Popular-Grade FPGA Devices. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 431–438. Springer, Heidelberg (2013)
14. Sugier, J.: Low-cost hardware implementation of Serpent cipher in programmable devices. In: Monographs of System Dependability: Technical Approach to Dependability, vol. 3, pp. 159–172. Publishing House of Wrocław University of Technology, Wrocław (2010)

Distributed Time Management in Wireless Sensor Networks

Tomasz Surmacz[1], Bartosz Wojciechowski[1], Maciej Nikodem[1], and Mariusz Słabicki[2]

[1] Institute of Computer Engineering, Control and Robotics, Wroclaw University of Technology, Poland
[2] Institute of Theoretical and Applied Informatics of Polish Academy of Sciences, Gliwice, Poland
tomasz.surmacz@pwr.wroc.pl

Abstract. Keeping time synchronization between nodes in wireless sensor networks (WSNs) is an important task allowing reduction of power consumption and better bandwidth usage due to collision avoidance. In this article we discuss requirements and practical limitations of time synchronization and present evaluation of sample algorithm, that can maintain the common clock between nodes of the WSN network that persists even in case of repeated node restarts, as long as network connectivity is maintained. We also provide measurements and temperature-dependent model of clock drift in popular WSN TelosB nodes.

1 Introduction

A common requirement for Wireless Sensor Network (WSN) nodes is the low cost and ability to operate for a long time without external power source. This allows setting up networks consisting of tens or hundreds of elements that can operate unattended for extensive periods of time, thus reducing the maintenance costs. For this reason WSN nodes are usually equipped with low-cost and low-power components and do not have any modules that can be considered superfluous, such as Real-Time Clocks (RTC) and high quality oscillators. This makes synchronization and time keeping in WSN nodes a challenge.

Agreeing on some common time in WSN nodes is needed for synchronous data transmission (TDMA modes), asynchronous sleep modes, collision avoidance, and data time-stamping, just to name a few uses. There are two closely-related topics: synchronization of the nodes and keeping real time. Most applications require only local time synchronization, i.e. between nodes in close vicinity of each other. This allows improving power conservation and communication throughput by synchronizing sleep/communication duty cycling and scheduling transmission times. In most cases the time between taking the measurement by a node and the time when the packet with measured quantity reaches the Base Station (BS) is short enough to time-stamp the measurement at it's destination. However, time-stamping the data at it's source may be necessary, for instance if some

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_43,

form of data aggregation is employed. This would allow for delayed transfer of non-time-critical data to reduce the total number of transmissions.

Many characteristics of WSN networks affect time-keeping and synchronization in the nodes. The most important ones are: clock frequency skews (between nodes), time drift (causing offsets), temperature-dependence of frequency of oscillators, random communication latencies and network topology variations, which may be caused by routing and network-management algorithms or by intrinsically low reliability of individual nodes causing failures and forcing changes in data transmission paths. Also in many low-power or high-throughput applications the amount of communication overhead associated with synchronization may be a problem.

Most nodes in a typical WSN for environment monitoring operate in similar ambient conditions. This causes their clocks to drift with similar pace. However, if some nodes' temperature differs from the others by a significant margin, e.g. due to heat build-up from direct sunlight, their clock drift may be much higher. In such cases these nodes may have a negative impact on time synchronization of the whole network. A time synchronization algorithm may use oscillator models to adjust the level of confidence of a given node.

This paper presents evaluation of a simple yet communication-effective approach to time-keeping and node-synchronization targeted at WSNs for environment monitoring. Such networks operate with long sleep periods to conserve energy. The time synchronization is only one of the tasks that the WSN network must perform – the main one (in our case) is performing periodical measurements and sending the results to BS. We focus on both extending the network lifetime and the reliability of data collection [1, 2].

Contributions of this paper are twofold: 1) evaluation of a simple world-time synchronization algorithm for environment monitoring WSNs using a real-world network deployed in a greenhouse, 2) measurement-based models of temperature-related differences in time drift of popular WSN nodes.

Our approach is to focus on collective time keeping in WSNs in cases where nodes may spontaneously reboot due to watchdog conditions caused by reliability requirements and need new time synchronization without any external time source. We show that even simple solutions to synchronization problems are satisfactory in environment monitoring applications while preserving low communications and energy overhead.

2 Related work

Most works focus on improving precision of time synchronisation. This usually comes at cost of multiple messages that have to be sent. One option is to transmit synchronization data by piggybacking with other types of packets, e.g. standard measurement reports. In most synchronization schemes it is necessary to be able to measure transmission delays between two points. The precision of message delay measurement depends on 2-way communication which is influenced by *send, access, transmission, propagation, reception* and *receive* times and is subject to high uncertainty. Even more so in WSNs with multi-hop communication

and frequent transmission collisions. The other important factor is the drift of nodes' internal clock [3]. For reasons such as computational requirements, memory footprint and communication bandwidth requirements, traditional approach to network synchronization known from TCP/IP networks, such as the Network Time Protocol (NTP) or IEEE 1588 Precision Time Protocol (PTP), are not suitable for use in low-communication-intensity, low-power sensor networks. Elson and Römer [4] explain why NTP is ill-suited for sensor networks and suggest various design principles for WSN time synchronization. They suggest adapting methods specific to the application and exploiting domain knowledge. The authors place a few known algorithms in the parameter space with *energy, precision, cost, synchronization scope* and *lifetime* as dimensions, but do not present any specific algorithm. In [5] Elson et al. presented Reference-Broadcast Synchronization (RBS) in which nodes send reference broadcasts to their neighbours to remove the sender's nondeterminism from timestamp calculation. The broadcasts are used as reference points for comparing clocks. The authors claim that this method improves precision with respect to two-point sender-receiver schemes. Cho at al. [6] consider using PTP for WSNs but their approach requires redesigning the hardware of a basic WSN node to contain a WSN to ethernet gateway and is not suitable for typical low-energy WSN networks.

Ganeriwal et al. proposed a two-phase hierarchical synchronization protocol called TPSN (Timing-Sync Protocol for Sensor Networks) [7]. In the first phase (*level discovery*) each node obtains a level, with only one *root node* having level 0. The *root node* is usually the BS (or packet sink) of the network. In the *synchronization* phase each node synchronizes to a node with a lower level. After the second phase of the protocol the network is globally synchronized to the *root node*. TPSN was implemented and tested on Berkeley's Mica nodes and allowed to achieve average error of less than $20\,\mu$s. Authors of [8] presented two lightweight synchronization algorithms called *tiny-sync* and *mini-sync*. These are based on the assumption that oscillators have fixed frequency and that two clocks can be linearly related. Then, a two-way communication is sufficient to obtain *data points* (3-tuples) that allow to bind the relative clock drifts and offsets of any pair of neighbouring nodes. LTS proposed by Greunen and Rabaey [9] takes a different approach aiming not at absolute accuracy but trying to minimise the overhead associated with synchronizing the nodes with required precision. This follows from the notion that the required accuracy in WSN networks is not very high (on the order of fractions of a second). TSync [10] follows a bidirectional approach where synchronization can be initiated by central time source (e.g. a node equipped with GPS receiver) for lightweight global synchronization or *pulled* on-demand by individual sensors.

In [11] a lightweight and energy efficient time synchronization scheme called LEETS is proposed. It can be applied to all kinds of TDMA power saving MAC schemes. The main objectives for LEETS design was to remove communication overhead associated with other time-synchronisation schemes. LEETS operates in two phases: initial time synchronisation and synchronisation maintaining, and assumes a root node with a GPS receiver to be present that sends the original

SYNC packet. Fontanelli and Macii [12] evaluate their *master-less local synchronization* algorithm by means of simulation. They assume pessimistic clock rate distribution over the total of 10^{-4} (100 ppm). These assumptions stand in contradiction to our own experimental evaluation of clock drift in WSN nodes (cf. Sec. 5).

A good introduction to clock synchronization can be found in a survey by Sundararaman et al. [13] where challenges and design principles relating strictly to WSN domain are presented and several algorithms are compared qualitatively and quantitatively.

3 Time Synchronization Approach

There are two main goals to achieve by using time synchronization: local synchronization within a group of nodes (which can span across the whole network) and synchronization with real-world time. Local synchronization provides means of scheduling sleep and alert modes at the same time at all nodes, so they may communicate efficiently and conserve power. Global network synchronization with universal time is not needed to achieve these goals, but on the other hand – is desirable for data timestamping. If nodes know the global time, they may locally optimize scheduling of data transmissions by aggregating several measurements over time and reducing the overall number of transmitted packets.

We have designed and implemented an algorithm (shown in Fig. 1.) which maintains the global time throughout the network with low message overhead. It combines reference broadcasts [5] with peer-to-peer node synchronization to achieve self-correcting behaviour of the WSN. In real implementations time synchronization is only one of the tasks that a WSN node must perform, so the algorithm is shown as a packet handling routine for received packets.

Initially, all network nodes start in unsynchronized state. Each node sends a time synchronization request just after booting, but these get unanswered as

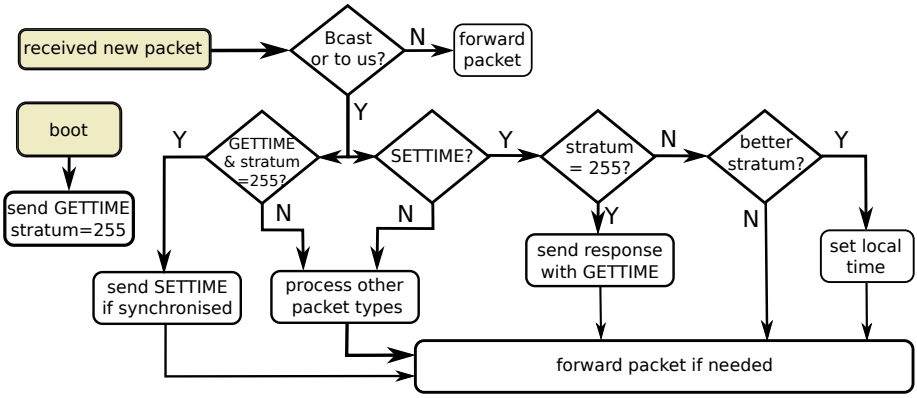


Fig. 1. Message handling in the implemented synchronization method

none of the network nodes running so far is able to answer that request. First synchronization is initiated by a broadcast of a `SETTIME` message from a network node maintaining the universal time reference. This is usually a Base Station but it may be also a wireless node with a Real Time Clock or GPS module attached. Once the message containing the global time reference is propagated throughout the whole network, all nodes change the state to synchronized.

From that point the whole network is synchronized and any new node appearing on the network will get its global synchronization from any other node in its vicinity. This will work properly if there is at least one synchronized node in the running network. Sync requests are broadcasted and retransmitted throughout the network. The distance from the time source measured in the number of retransmissions of the sync packet is called *stratum* – when setting local time, messages with lower stratum are preferred (stratum 0 meaning the clock source, stratum 1: one hop from the source, and so on). Lets assume a sync request message is sent at time t_0 from node X and received at t_1 at node Y. Before retransmission, node Y checks its local time and if the node is in synchronized state, then a response message is sent (at time t_2) which will be eventually delivered to node X at time t_3. Message processing time $(t_2 - t_1)$ is marginal (usually constant). The accuracy of time extracted at X is influenced mostly by message propagation delay $(t_3 - t_2)$, which increases with the number of retransmissions. To compensate for that, time at X is set to $t_2 + \frac{t_3 - t_0}{2}$.

Each node maintains its local time by examining a millisecond resolution timer which is started at system bootup, thus reflecting the node uptime. Time synchronization status is kept in 3 variables: time sync source, *stratum* which refers to the clock source distance, and the millisecond accuracy time difference between the real time and local uptime clock. Initially, *stratum* is set to 255, which means that the real time is unknown.

Global time synchronization is acquired by propagating a time setup message (`SETTIME`) throughout the network. Each node that receives such a message compares message's *stratum* and *noOfHops* data with the local `stratum` parameter and if $msg\_stratum + noOfHops < local\_stratum$, it stores the difference between the received and local time. If the time setting message was a broadcast message, it is resent to other nodes (and the `noOfHops` increases on each retransmission).

Base Station (or any node) may request other node's time info by sending `SETTIME` message with `stratum` parameter set to 255. The responding node replies with a `GETTIME` message (using the same message layout) containing its own source reference, stratum and freshly calculated real time. A `GETTIME` with `stratum` equal to 255 heard at other nodes triggers sending a `SETTIME` with local time info, which allows for automatic self-correction of "lost in time" nodes – i.e. nodes that overhear "unsynchronized" response automatically respond with their own sync data.

The algorithm has been implemented and tested in TinyOS running on TelosB and XM1000 motes.

4 Test Cases

The algorithm has been tested in real-life network setup in greenhouse environment monitoring application. Our test network is based on popular TelosB motes and programmed with TinyOS. For the sake of simplicity and reliability a multi-layered protocol is used that can either build a routing tree between WSN nodes or use message flooding with broadcasts. Each message is identified by a source node and a sequence number, which are preserved if the message is retransmitted. All nodes keep track of *(source, seq)* pairs of messages seen from their neighbours, so that transmission loops and retransmissions of duplicates are avoided. We use the standard Low Power Listening mechanism to prolong nodes' lifetime through radio duty-cycling. Eleven wireless nodes and two Base Stations have been placed in total throughout 3 greenhouse buildings.

Both Base Stations were used for data gathering during the experiment and only one initial synchronization message has been sent at the beginning. After that, all nodes have been using only their own clocks to maintain the global time reference. The data shown below describes the experiment that started on November 13th 2013 and lasted until Nov 28th (over 2 weeks). As can be seen in Fig. 2, during this test nodes steadily gained clock drift between 2 and 3 seconds from the universal time, which increased to 5-7 s in the last 2 days when network connectivity detoriated rapidly. The relative difference between all nodes was kept within fractions of a second. For the clarity of pictures not all nodes are shown, but the omitted ones present the same data patterns.

Nodes were programmed with watchdog features that rebooted each node if any malfunction was detected, such as a stalled timer, buffer overrun, etc. In normal mode of operation such reboots would occur at a much lesser rate, but for the sake of timekeeping experiments this was the desired situation, allowing

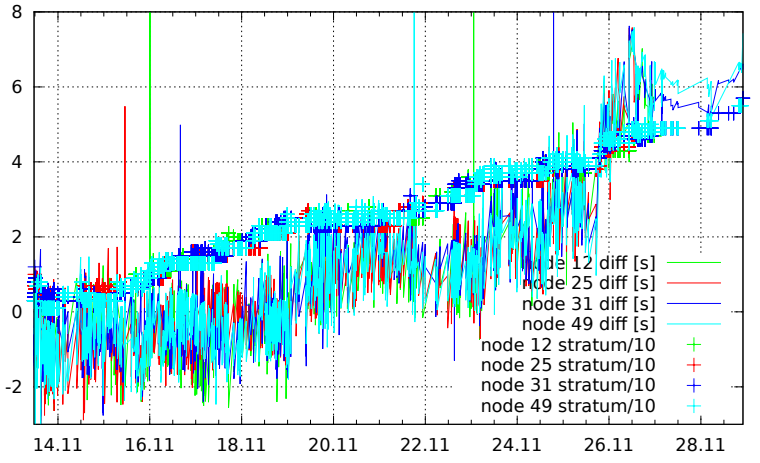


Fig. 2. Stratum and time drift experiment data

```
1  2013-11-13 12:22:30.2638, from=79, to=65535, via=79, seq=1,  hops=1,
       msg_t=GETTIME, stratum=255, rtc=1970-01-01 01:00:00.331000, rtc_s=0.331,
2  2013-11-13 12:22:30.2812, from=32, to=65535, via=32, seq=56, hops=1, msg_t=
       SETTIME, stratum=6, rtc=2013-11-13 12:22:30.272, rtc_s=1384341750.272,

3  2013-11-13 12:22:30.3425, from=31, to=65535, via=31, seq=29, hops=1, msg_t=
       SETTIME, stratum=5, rtc=2013-11-13 12:22:31.466, rtc_s=1384341751.466,

4  2013-11-13 12:22:30.3595, from=31, to=65535, via=82, seq=29, hops=2, msg_t=
       SETTIME, stratum=5, rtc=2013-11-13 12:22:31.466, rtc_s=1384341751.466,

5  2013-11-13 12:22:30.3932, from=31, to=65535, via=5,  seq=29, hops=2, msg_t=
       SETTIME, stratum=5, rtc=2013-11-13 12:22:31.466, rtc_s=1384341751.466,

6  2013-11-13 12:22:30.3934, from=31, to=65535, via=49, seq=29, hops=2, msg_t=
       SETTIME, stratum=5, rtc=2013-11-13 12:22:31.466, rtc_s=1384341751.466,

7  2013-11-13 12:22:30.4192, from=31, to=65535, via=25, seq=29, hops=2, msg_t=
       SETTIME, stratum=5, rtc=2013-11-13 12:22:31.466, rtc_s=1384341751.466,

8  ...
9  2013-11-13 12:26:19.1149, from=22, to=65535, via=22, seq=1, hops=1, msg_t=
       GETTIME, stratum=255, rtc=1970-01-01 01:00:00.337, rtc_s=0.337,
10 2013-11-13 12:26:19.2269, from=79, to=65535, via=79, seq=6, hops=1, msg_t=
       SETTIME, stratum=6, rtc=2013-11-13 12:26:19.144, rtc_s=1384341979.144,
```

Fig. 3. Sample time sync exchange between nodes

us to test the algorithm performance in extreme conditions. The increasing value of stratum in all nodes indicates repeating node reboots and slowly deteriorating base time reference.

Fig. 3 shows a sample message exchange in timesync protocol. First, node 79 reboots and sends its time request message (noted by seq=1 – see line 1). Then node 32 responds with its own stratum 6 data (line 2) and node 31 with stratum 5 (line 3) which is also received as a retransmitted packet later (lines 4, 5, 6 and 7). Later on, when node 22 reboots (line 9) node 79 is one of those responding with its own time (line 10) referenced with stratum 6. This means that after initial setup from node 32 to stratum 7 it readjusted its time info to stratum 6 after receiving the stratum 5 message shown in line 3. Further messages did not improve the time sync data, as their $stratum + hops$ values exceeded the already set value.

Although the final results of 5 seconds timedrift in 2 weeks may not look astonishing, it is actually a $3.9 \cdot 10^{-6}$ precision for collective network timekeeping without any external time reference (except the initial time setting). This has to be compared to the quality of crystal oscillators available in WSN nodes. To keep WSN node costs within reasonable bounds these are just popular electronics quality parts. We have performed several tests to determine the clock drift behaviour of these devices and the results are shown in Fig. 4. In these experiments each node was running a simple program that periodically sent the value of its free running uptime counter, thus informing of its local time. Packets from all nodes were received directly by a Base Station located nearby (no retransmissions). Each logged packet contained three timestamps: the uptime counter

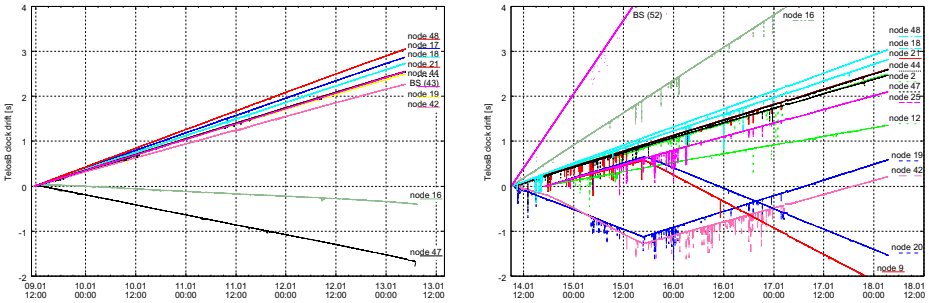


Fig. 4. Clock drift comparison of TelosB nodes in 2 experiments

of the originating node, Base Station's timestamp added as soon as the packet has been received over the radio and queued over the serial interface to the NetServ [2] program running on a Raspberry Pi computer, and finally – a Netserv's timestamp of a packet received over the serial line. The first two are relative to when a particular WSN node was booted, while the last one is a NTP-based accurate global time. Thus, an increasing or decreasing difference in node's and NetServ's timestamps measures the clock drift of a particular WSN node.

Nodes 16 and 47 in first experiment (Fig. 4 left) showing a negative clock drift were placed in a cooled environment (around 5°C), while all others were placed in room temperature of 22°C. The second experiment (Fig. 4 right) started with nodes 19 and 42 placed in the cooled environment, but these have been replaced with nodes 9 and 20 at the 15.01@16:00 mark on the timescale. The effect of this replacement can be easily seen on the corresponding timedrift measurements. Momentary negative drifts that can be seen on Fig. 4 (right) until 17.01 03:00 are the effect of the slightly overloaded BS system (Raspbian Linux on Raspberry Pi) which ended when system was rebooted due to the mains power failure.

As can be seen, the drift ratio of all nodes falls around 1s/day (i.e. 10^{-5} clock precision), but is also node-specific and depends heavily on ambient temperature [14]. In order to achieve a better timekeeping precision of WSN nodes two factors would have to be considered: an individual node clock calibration data and temperature compensation. The first one has to be determined individually for each node before the network deployment while the second requires constant temperature monitoring of the node and applying appropriate corrections. One of the options would be to use TCXOs (Temperature Compensation Crystal Oscillators) instead of SPXOs (Simple Packaged Crystal Oscillators) in node design but that would increase the node cost while improving the oscillator accuracy from 10^{-5} to 10^{-6} ppm. If frequent temperature measurements are the core functionality of the WSN network, these can be used for time corrections at each node (i.e. microcontroller-driven time compensation), but if the measurements are rare and the nodes spend most of the time in sleep modes to conserve the batteries, these extra wakeup states would not be justified.

5 Clock Drift Measurements

To find out how the clock drift changes with temperature we performed measurements of both quantities in outdoor conditions, during a week when outside temperatures steadily dropped over 15 degrees from 6 to −9 °C and in indoor/lab controlled conditions of temperatures reaching 60°C. For this experiment we used three TelosB nodes (10, 15 and 45) and two XM1000 nodes (52, 53). We averaged temperature every $dt = 900$ s and plotted time drift for each time period as a function of temperature (see Fig. 5). We then fitted parameters of polynomial functions to all the data sets. According to [14] the crystal oscillator's drift can be fitted by a cubic function. Our measurements show that for a microcontroller clock drift a square function is enough. Using higher-order polynomials did not improve the fit quality by a substantial value (measured by the norm of residuals). For example, for node 52 (XM1000 architecture) the fitted functions are $-0.036x^2+1.7x+16$ and $-0.000034x^3-0.033x^2+1.7x+16$ respectively, and for node 45 (TelosB): $-0.037x^2+1.7x-11$ and $-0.000011x^3-0.036x^2+1.7x-11$.

The drift data shown in Fig. 5 is subject to noise. This is mainly caused by random noise originating from delays in interrupt handling in the nodes and transmission delays caused by collisions (10ms drift measurement error over 900s measurement period results in a 11 ppm error). However, removing the noise by applying median filtering did not change the relation between operating temperature and average time drift of the oscillator. For all the nodes the maximal drift change is close to 1.8 ppm per K. TelosB nodes have the drift close to 0 at around 6 °C and 40 °C while the XM1000 nodes have minimal clock drift around −3 °C and 57 °C.

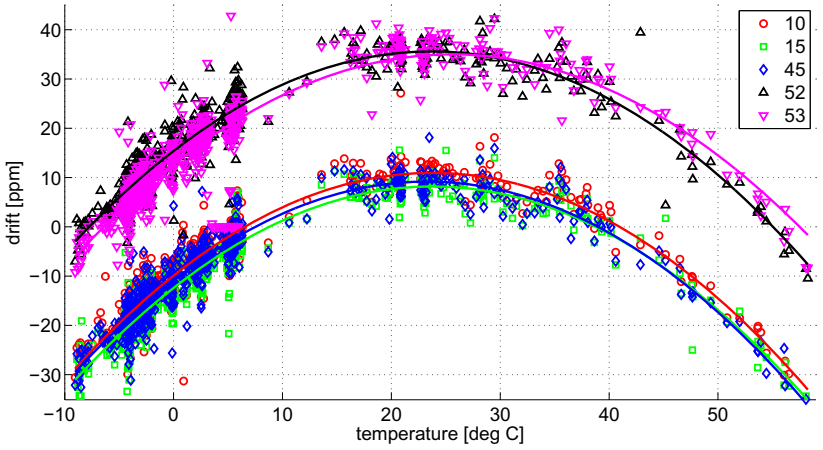


Fig. 5. Clock drift of nodes as a function of temperature

6 Conclusions

Time synchronization is an important issue in WSNs as it allows to reduce energy consumption by scheduling transmissions and sleep/awake duty cycling. Practical experiments show that using even simple synchronization methods without the external time source allows to maintain the global network time drift below $5 \cdot 10^{-6}$ which is comparable to the accuracy of the precision quartz crystal oscillators. For synchronous sleep modes this is acceptable, as the algorithms provide self-adjustment to minor discrepancies. This level of precision is also sufficient in data gathering applications for environment monitoring. We have also measured the characteristics of clock drift in function of temperature for TelosB and XM1000 nodes. These can be described by simple quadratic functions, so if periodic temperature measurements are taken by WSN nodes in their typical mode of operation, they can be used to calculate appropriate clock drift compensations.

Acknowledgement. This work was supported by National Science Centre grant no. N 516 483740.

References

1. Surmacz, T., Słabicki, M., Wojciechowski, B., Nikodem, M.: Lessons learned from the deployment of wireless sensor networks. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2013. CCIS, vol. 370, pp. 76–85. Springer, Heidelberg (2013)
2. Surmacz, T.: Rapid protocol development in wireless sensor networks using wireshark plugins. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2013. LNCS, vol. 8112, pp. 426–433. Springer, Heidelberg (2013)
3. Ageev, A., Macii, D., Petri, D.: Synchronization uncertainty contributions in wireless sensor networks. In: Instrumentation and Measurement Technology Conference Proceedings, IMTC 2008, pp. 1986–1991. IEEE (2008)
4. Elson, J., Römer, K.: Wireless sensor networks: A new regime for time synchronization. ACM SIGCOMM Computer Communication Review 33(1), 149–154 (2003)
5. Elson, J., Girod, L., Estrin, D.: Fine-grained network time synchronization using reference broadcasts. ACM SIGOPS Operating Systems Review 36(SI), 147–163 (2002)
6. Cho, H., Son, S., Baek, Y.: Implementation of a precision time protocol over low rate wireless personal area networks. In: 13th Asia-Pacific Computer Systems Architecture Conference, ACSAC 2008, pp. 1–8 (August 2008)
7. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-sync protocol for sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 138–149. ACM (2003)
8. Sichitiu, M., Veerarittiphan, C.: Simple, accurate time synchronization for wireless sensor networks. In: 2003 IEEE Wireless Communications and Networking, WCNC 2003, vol. 2, pp. 1266–1273 (2003)
9. Van Greunen, J., Rabaey, J.: Lightweight time synchronization for sensor networks. In: Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, pp. 11–19. ACM (2003)

10. Dai, H., Han, R.: TSync: a lightweight bidirectional time synchronization service for wireless sensor networks. ACM SIGMOBILE Mobile Computing and Communications Review 8(1), 125–139 (2004)
11. Xu, M., Zhao, M., Li, S.: Lightweight and energy efficient time synchronization for sensor network. In: Proceedings of the 2005 International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, pp. 947–950 (2005)
12. Fontanelli, D., Macii, D.: Master-less time synchronization for wireless sensor networks with generic topology. In: 2012 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 2785–2790 (2012)
13. Sundararaman, B., Buy, U., Kshemkalyani, A.D.: Clock synchronization for wireless sensor networks: A survey. Ad Hoc Networks 3(3), 281–323 (2005)
14. Zhou, H., Nicholls, C., Kunz, T., Schwartz, H.: Frequency accuracy & stability dependencies of crystal oscillators. Carleton University, Systems and Computer Engineering, Technical Report SCE-08-12 (2008)

Heuristic Cycle-Based Scheduling with Backfilling for Large-Scale Distributed Environments

Victor Toporkov[1], Anna Toporkova[2],
Alexey Tselishchev[3], Dmitry Yemelyanov[1], and Petr Potekhin[1]

[1] National Research University "MPEI",
ul. Krasnokazarmennaya, 14, Moscow, 111250, Russia
{ToporkovVV,YemelyanovDM,PotekhinPA}@mpei.ru
[2] National Research University Higher School of Economics,
Moscow State Institute of Electronics and Mathematics,
Bolshoy Trekhsvyatitelsky per., 1-3/12, Moscow, 109028, Russia
atoporkova@hse.ru
[3] European Organization for Nuclear Research (CERN),
Geneva, 23, 1211, Switzerland
Alexey.Tselishchev@cern.ch

Abstract. The paper is devoted to comparing the results of an independent job batch scheduling in terms of a virtual organization policy and available resources usage efficiency in large distributed environments like utility Grid. A hybrid approach is proposed on the basis of a cyclic scheduling scheme and backfilling combination. Additionally the paper offers a heuristic shifting procedure which improves jobs execution alternatives selected in the cyclic scheme. The simulation results show that depending on the scheduling efficiency indicator and the level of resource availability each of the approaches is able to provide the best results. Moreover the obtained results are valid under conditions of dynamically varying state of resources and inaccurate user job runtime estimations.

Keywords: Distributed computing, economic scheduling, slot, job, backfilling.

1 Introduction

Some of the most important quality of service (QoS) indicators of a distributed computational environment is a utilization level of the available resources and job start ("response") time. In distributed environments with non-dedicated resources the computational nodes are usually partly utilized by local priority jobs. Thus, the resources available for use can be represented as a set of slots – time intervals during which the individual computational nodes are vacant to execute parts of multiprocessor parallel jobs. Presence of this set of slots (which generally have different start and finish times and difference in performance depending on the node, where the slot is allocated) impedes the problem of a coordinated selection of the resources required to execute the job flow from the computational environment users.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_44,

Resource fragmentation also results in a steady decrease of the total computing environment utilization level. Resource management and job scheduling economic models proved to be efficient in such conditions [1-8]. Application-level scheduling, as a rule, does not imply any global resource sharing or allocation policy. Resource brokers [9-15] are usually considered as mediators between users and resource owners. Scheduling and resource management systems in this approach are well-scalable and application-oriented. However, simultaneous application-level scheduling with diverse optimization criteria set by independent users, especially upon possible competition between applications, may deteriorate such QoS characteristics of a distributed environment as total job batch execution time or overall resource utilization. The regulations of a virtual organization (VO) in Grid [16] usually suppose a job flow scheduling. A meta-scheduler or a meta-broker are considered as intermediate chains between the users and a local resource management and job batch processing systems [1, 17-20]. VOs, on one hand, naturally restrict the scalability of resource management systems (though, it is worth remarking here, that there is a good experience [19] of enabling interoperability among meta-schedulers belonging to different VOs). On the other hand, uniform rules of a resource sharing and consumption, in particular based on economic models [1-8], make it possible to improve the job-flow level scheduling and resource distribution efficiency. In some well-known models of a distributed computing environments with non-dedicated resources, only the first fit set of resources is chosen depending on the environment state [18, 21-23], while job scheduling optimization mechanisms are usually not supported. In other models [2, 3, 17] the aspects related to the specifics of the environments with non-dedicated resources (particularly dynamic resource loading, the competition between independent users, users' global and owners' local job flows) are not presented.

In this paper, we propose a combined approach to meta-scheduling in VOs. First of all, we address a problem of an early resources release and "on the fly" rescheduling by combining our original cyclic scheduling scheme (CSS) [24] with backfilling [25]. For overall job-flow execution optimization and a resource occupation time prediction existing schedulers rely on the time specified in the job request, e.g. using Job Submission Description Language (JSDL). However, the reservation time is usually based on the user inaccurate runtime estimates [26]. In case, when the application is completed before the term specified in the resource request, the allocated resources remain underutilized. Second, we introduce a schedule shifting heuristic in CSS. Thus, we outline two main job-flow optimization directions. First, optimal or a suboptimal (under a given VO criteria) scheduling is performed on the basis of a priori information about the computational nodes local schedules and the resource reservation time for each job execution. CSS belongs to this type of systems. Another approach represents scheduling "on the fly" which depends on dynamically updated information about resource utilization. In this case, schedulers are focused on overall resources load maximization and a job start time minimizing. Backfilling may be related to this scheduling model.

The rest of the paper is organized as follows. Section 2 is devoted to analysis of the related works. In Section 3, there is a concept of CSS and backfilling combination as well as of the shifting procedure. Section 4 contains simulation results of the considered scheduling approaches comparison. Finally, section 5 summarizes the paper and describes further research topics.

2 Related Works

Many resource selection and scheduling algorithms, and heuristic-based solutions have been proposed for parallel jobs and tasks with dependencies in distributed environments [3, 18, 19, 21-23, 27-33].

In [3], heuristic algorithms for slot selection, based on user-defined utility functions, are introduced. Slot window allocation is based on the user defined efficiency criterion under the maximum total execution cost constraint. However, the optimization occurs only on the stage of the best found offer selection. The paper [28] presents architecture and an algorithm for performing Grid resources co-allocation without the need for advance reservations based on synchronous subtasks queuing. However, advance reservation is efficient to improve the co-allocation QoS. Advance reservation-based co-allocation algorithms are proposed in [21-23, 29, 30].

First fit resource selection algorithms [21-23] assign any job to the first set of slots matching the resource request conditions without any optimization. Preference-based matchmaking [18] is not focused on the scheduling process. The job is scheduled on the first available resource according to user preferences. In [19], an approach to resource matchmaking among VOs combining hierarchical and peer-to-peer meta-schedulers models is proposed.

The co-allocation algorithms described in [29-31] suppose an exhaustive search and some of them are based on a linear integer programming (IP) [7, 30] or a mixed-integer programming (MIP) model [31]. The co-allocation algorithm presented in [30] uses the 0-1 IP model with the goal of creating reservation plans satisfying user resource requirements. Users can specify a time frame for each resource: the earliest start time, the latest start time and the job duration, where user wants to reserve a time slot. This condition imposes restrictions for slots search only within this time frame. A linear IP-driven algorithm is proposed in [7]. It combines the capabilities of an IP and a genetic algorithm and allows obtaining the best meta-schedule that minimizes the combined cost of all independent users in a coordinated manner. In [31], the authors propose a MIP model which determines the best scheduling for all the jobs in the queue in environments composed of multiple clusters that act collaboratively. The scheduling techniques proposed in [7, 29-33] are efficient compared with other scheduling techniques under given criteria: the minimum processing cost, the overall makespan, resources utilization, load balancing for related tasks [32, 33], etc. However, complexity of the scheduling process is extremely increased by the resources heterogeneity and the co-allocation process, which distributes the tasks of parallel jobs across resource domain boundaries.

In this work, we use algorithms for efficient slot selection based on criteria defined by users, resource owners and VO administrators. The algorithms have linear complexity against the number of all available time-slots and operate on a scheduling interval denoting how far in the future the system may schedule resources [24, 27].

3 A Concept of Combined Cycle-Based Scheduling

CSS was proposed for a model based on a hierarchical job-flow management [24]. Job-flow scheduling is performed in cycles with separate job batches on the basis of dynamically updated computational nodes' local schedules.

Among the major CSS restrictions in terms of an efficient scheduling and resource allocation one may outline the following. First of all, it is not possible to affect execution parameters of an individual job: the search for particular alternatives is performed on the First Fit principle, while choice of the optimal alternatives combination represents only the VO interests. Second, the job batch scheduling is based on an often inaccurate user estimation of a particular job runtime [26]. Third, the job batch scheduling requires allocation of a multiple "nonintersecting" in terms of slots alternatives, and only one alternative is chosen for each job execution.

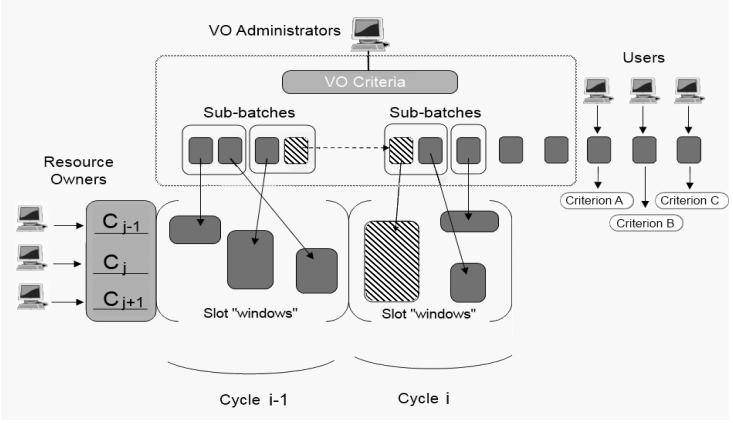


Fig. 1. Cyclic scheduling with batch-slicing

We introduce a modified CSS model: Batch-slicer (Fig. 1). In order to satisfy the user preferences a desirable optimization criterion is introduced into the job request. In Fig. 1: Cj-1, Cj, Cj+1 are slot costs determined by resource owners. We propose initial job batch separation into a set of sub-batches and each sub-batch scheduling at the same given scheduling interval. According to the alternatives search algorithm adopted in CSS [24], at a high resource utilization level the number of the batch jobs' execution alternatives may be relatively small up to just a single alternative for every job. Such a small number of alternatives found may affect the optimal slot

combination selection, and therefore, may reduce overall scheduling efficiency. The job batch "slicing" increases the number of alternatives found for high-priority jobs and diversifies the choice on the slots combination selection stage, and thereby increases the resource sharing efficiency according to VO policy.

Backfilling [25] responds to early resources releases and performs "on the fly" rescheduling which is very important when a user job runtime estimation is significantly different from the actual job execution time. However backfilling has some limitations for distributed computing. The first one is inefficient resource usage by criteria different from average job start time (especially at a relatively low resource load level). The second one is a principal inability to affect the resource sharing quality by defining policies and criteria in VO.

We introduce a combined approach. During every scheduling cycle a set of high priority jobs is allocated from the initial job batch. These jobs are grouped into a separate sub-batch and should be scheduled before other jobs, probably, without compliance with the queue discipline. The scheduling of this sub-batch is further performed by Batch-slicer based on the preliminary known resources utilization schedule. The scheduling of the rest batch jobs is performed by backfilling with the dynamically updated information about the actual computational nodes utilization. The cyclic scheduling method combined with backfilling (Batch-slice-Filling - BSF) combines the main advantages of both Batch-slicer and backfilling, namely the optimization of the most time-consuming jobs execution as well as the efficient resource usage, preferential job execution queue order compliance and a relatively low response time.

A heuristic shifting procedure is proposed for the job execution alternatives selected for advanced reservation. The procedure shifts the alternatives in time towards the beginning of the scheduling interval retaining resource instances in which they are allocated. The shifting procedure is done iteratively for each job of the batch being scheduled. The job selection order is determined according to the start time of the chosen alternatives: first, an attempt to shift the alternatives with the minimal start time is performed. Such order guarantees that when shifting a job all other jobs with an earlier start time are already shifted and hence do not occupy the corresponding nodes. Otherwise, a task with an earlier start time and a lower priority may block the shift of a task with a higher priority and then, in its turn, may be shifted releasing extra slots.

4 Simulation Studies

The experiments are devoted to study scheduling efficiency using the proposed approaches: CSS, BSF, Shifted CSS (CSS with the use of a shifting procedure), and also backfilling (BF). The goal is to compare the scheduling efficiency depending on the number of computational nodes in the domain as well as to investigate schedules

consistency under conditions of inaccurate user job execution time estimations. A series of studies were carried out with the simulation environment [24]. Each experiment includes an input batch of 15 jobs generation as well as the resources structure and local schedules of the computational environment. To analyze the approaches under different conditions the simulation is conducted individually for different numbers of the nodes available {6, 10, 20, 30, 50, 75, 100, 150}. Thus the investigation consists in comparing the scheduling results obtained with the same input data by means of different algorithms.

Scheduling efficiency is considered from the viewpoint of a job batch total slot utilization time T_{proc} minimization, start t_{start} and finish t_{finish} job batch execution times minimization, and minimization of a combined criterion $F = t_{start} + T_{proc}$. For a batch consisting of multiple jobs we consider average parameter values.

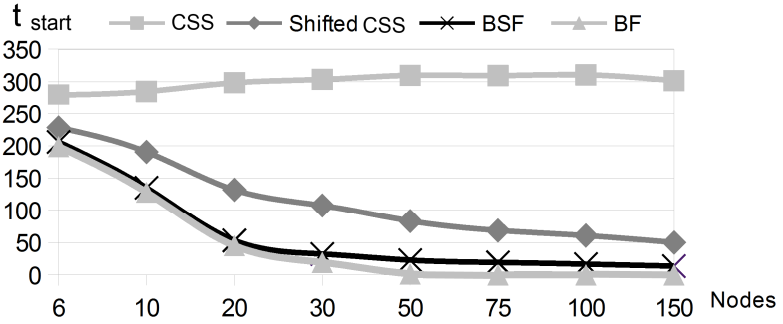


Fig. 2. Average job batch start time

Figure 2 shows average scheduled job start times obtained independently with all considered scheduling approaches depending on the computational environment nodes number. As can be seen from Fig. 2, with increasing amount of available computational nodes backfilling is able to reduce the average job batch start time down to zero (i.e. all batch jobs can start at the very beginning of the scheduling interval). At the same time average job start time obtained with CSS is almost independent of the available nodes number. BSF provides the average job batch start time close to backfilling's by *filling* unused by CSS time slots near beginning of the scheduling interval with relatively low priority jobs. With a relatively large resource level an average job batch start time obtained with Shifted CSS tends to a non-zero value since the most profitable in terms of the optimization criterion resources are generally allocated for more than one job. Thus in case of heterogeneous resource environment it is virtually impossible to start all the batch jobs at the beginning of the scheduling interval using CSS (even with *shifted* variation).

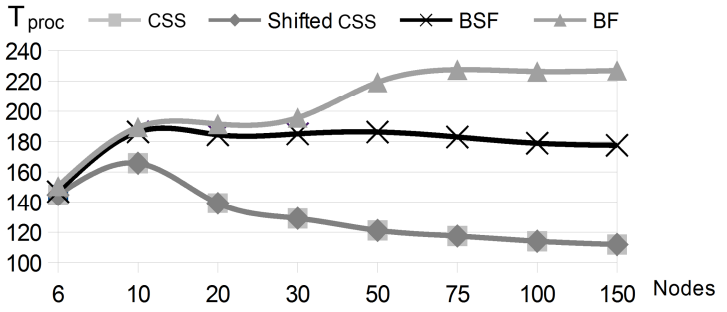


Fig. 3. Average batch jobs processor time usage

Figure 3 shows the advantage of CSS, Shifted CSS, and BSF over backfilling by the VO target optimization criterion T_{proc}. It should be noted that with increasing number of available resources the advantage of CSS and BSF over backfilling also increases. The use of additional heuristics, such as job batch slicing, can provide even greater CSS advantage on the target criterion.

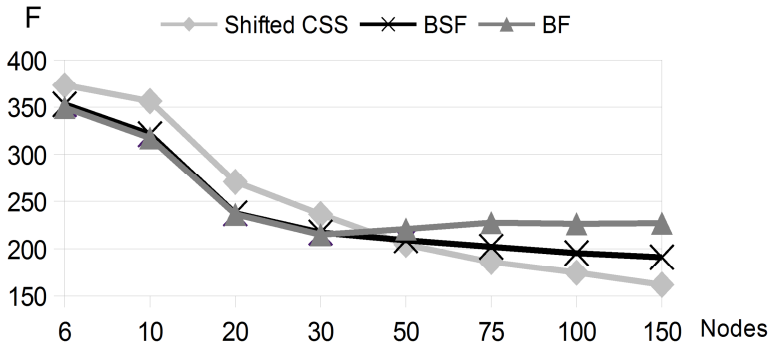


Fig. 4. Average batch jobs F value

Figure 4 shows the value of the combined resource usage efficiency index $F = t_{start} + T_{proc}$. It is important to note that the intersection between the Shifted CSS, BSF and backfilling graphs implies that in case of a relatively low level of available resources backfilling or BSF (they provide almost the same criterion F value) are better as compared with Shifted CSS. With increasing computational environment size Shifted CSS becomes more advantageous in terms of resource usage efficiency.

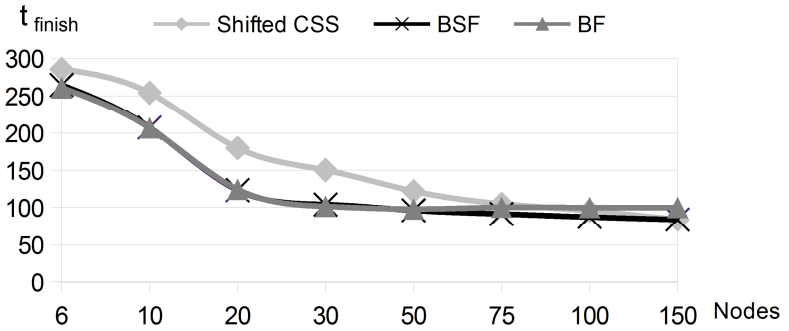


Fig. 5. Average batch jobs finish time

The same conclusion can be drawn if we evaluate the resource usage efficiency by the average batch job finish time (Fig. 5): Shifted CSS provides the best results when computational nodes with a sufficiently large number are available. Thus the use of BSF is justified in virtually any conditions: this combined approach provides competitive to backfilling values of all considered resource usage efficiency indexes, and at the same time optimizes execution performance of the high-priority jobs.

Another experiment studies the scheduling efficiency and consistency when based on user estimated job runtimes in case when these estimations are inaccurate. During the computational environment simulation batch jobs' actual execution time was set as a random variable with uniform distribution, which allows actual job execution time and a user estimation vary by 5 times. Uniform distribution is chosen as it is almost impossible to predict real job execution time on the specified resources. Table 1 contains the simulation results. Results show, that even if the difference between the resource reservation time and the real job execution time is significant the advantage of CSS over backfilling against the VO target optimization criterion not only remains but increases. That is because backfilling does not optimize against criteria different from the start time and implied more compact job packing uses almost all the available resources including those less advantageous against the target criterion.

Table 1. Average batch jobs processor time usage

Approach	Processor time usage	
	Reserved	Real
Backfilling	208	140
CSS	168	112
CSS advantage	19%	20%

5 Conclusions and Future Work

In this work, we compare the scheduling results of a batch of independent jobs in terms of a virtual organization policy and the available resources usage efficiency. Based on a cyclic scheduling scheme and backfilling combination a hybrid approach BSF is proposed. Additionally the shifting procedure is proposed for the alternatives chosen in CSS. The simulation results show that depending on the considered scheduling efficiency index, and depending on the level of the resources available, each of the considered approaches may provide the best results. Backfilling, as a rule, minimizes job start and finish times, while CSS is able, for example, to minimize the job processor time usage (when given the appropriate optimization criterion). In order to ensure compromise scheduling results it is justified to use BSF: scheduling of high priority jobs with CSS and further filling the remaining unassigned resources with backfilling. The results obtained remain valid in a dynamically changing computational environment condition and composition, and in case when user jobs runtime estimations are significantly inaccurate.

Further research will be related to a more precise investigation of dividing the job flow into sub-batches depending on the jobs characteristics and computational environment parameters as well as to studying of rescheduling based on the information about computational nodes current state and performance.

Acknowledgements. This work was partially supported by the Council on Grants of the President of the Russian Federation for State Support of Leading Scientific Schools (SS-362.2014.9), the Russian Foundation for Basic Research (grant no. 12-07-00042).

References

1. Garg, S.K., Konugurthi, P., Buyya, R.: A Linear Programming-driven Genetic Algorithm for Meta-scheduling on Utility Grids. J. Par., Emergent and Distr. Systems 26, 493–517 (2011)
2. Buyya, R., Abramson, D., Giddy, J.: Economic Models for Resource Management and Scheduling in Grid Computing. J. Concurrency and Computation 14(5), 1507–1542 (2002)
3. Ernemann, C., Hamscher, V., Yahyapour, R.: Economic Scheduling in Grid Computing. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2002. LNCS, vol. 2537, pp. 128–152. Springer, Heidelberg (2002)
4. Lee, Y.C., Wang, C., Zomaya, A.Y., Zhou, B.B.: Profit-driven Scheduling for Cloud Services with Data Access Awareness. J. Par. and Distr. Computing 72(4), 591–602 (2012)
5. Garg, S.K., Buyya, R., Siegel, H.J.: Scheduling Parallel Applications on Utility Grids: Time and Cost Trade-off Management. In: 32nd Australasian Computer Science Conference (ACSC 2009), Wellington, New Zealand, pp. 151–159 (2009)
6. Degabriele, J.P., Pym, D.: Economic Aspects of a Utility Computing Service, Trusted Systems Laboratory HP Laboratories Bristol HPL-2007-101. Technical Report, pp. 1-23 (July 3, 2007)

7. Garg, S.K., Yeo, C.S., Anandasivam, A., Buyya, R.: Environment-conscious Scheduling of HPC Applications on Distributed Cloud-oriented Data Centers. J. Parallel and Distributed Computing 71(6), 732–749 (2011)
8. Tesauro, G., Bredin, J.L.: Strategic Sequential Bidding in Auctions Using Dynamic Programming. In: 1st International Joint Conference on Autonomous Agents and Multiagent Systems, Part 2, pp. 591–598. ACM, New York (2002)
9. Thain, D., Tannenbaum, T., Livny, M.: Distributed Computing in Practice: the Condor Experience. J. Concurrency and Computation: Practice and Experience 17(2-4), 323–356 (2004)
10. Berman, F.: High-performance Schedulers. In: Foster, I., Kesselman, C. (eds.) The Grid: Blueprint for a New Computing Infrastructure, pp. 279–309. Morgan Kaufmann, San Francisco (1999)
11. Yang, Y., Raadt, K., Casanova, H.: Multiround Algorithms for Scheduling Divisible Loads. IEEE Trans. Parallel and Distributed Systems 16(8), 1092–1102 (2005)
12. Natrajan, A., Humphrey, M.A., Grimshaw, A.S.: Grid Resource Management in Legion. In: Nabrzyski, J., Schopf, J.M., Weglarz, J. (eds.) Grid Resource Management. State of the Art and Future Trends, pp. 145–160. Kluwer Academic Publishers, Boston (2003)
13. Beiriger, J., Johnson, W., Bivens, H.: Constructing the ASCI Grid. In: 9th IEEE Symposium on High Performance Distributed Computing, pp. 193–200. IEEE Press, New York (2000)
14. Frey, J., Foster, I., Livny, M.: Condor-G: a Computation Management Agent for Multi-institutional Grids. In: 10th International Symposium on High-Performance Distributed Computing, pp. 55–66. IEEE Press, New York (2001)
15. Abramson, D., Giddy, J., Kotler, L.: High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid? In: International Parallel and Distributed Processing Symposium, pp. 520–528. IEEE Press, New York (2000)
16. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int. J. of High Performance Computing Applications 15(3), 200–222 (2001)
17. Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J.: Multicriteria Aspects of Grid Resource Management. In: Nabrzyski, J., Schopf, J.M., Weglarz, J. (eds.) Grid Resource Management. State of the Art and Future Trends, pp. 271–293. Kluwer Academic Publishers, Boston (2003)
18. Cafaro, M., Mirto, M., Aloisio, G.: Preference-Based Matchmaking of Grid Resources with CP-Nets. J. Grid Computing 11(2), 211–237 (2013)
19. Rodero, I., Villegas, D., Bobroff, N., Liu, Y., Fong, L., Sadjadi, S.M.: Enabling Interoperability among Grid Meta-Schedulers. J. Grid Computing 11(2), 311–336 (2013)
20. Toporkov, V.: Application-Level and Job-Flow Scheduling: an Approach for Achieving Quality of Service in Distributed Computing. In: Malyshkin, V. (ed.) PaCT 2009. LNCS, vol. 5698, pp. 350–359. Springer, Heidelberg (2009)
21. Aida, K., Casanova, H.: Scheduling Mixed-parallel Applications with Advance Reservations. In: 17th IEEE Int. Symposium on HPDC, pp. 65–74. IEEE CS Press, New York (2008)
22. Ando, S., Aida, K.: Evaluation of Scheduling Algorithms for Advance Reservations. Information Processing Society of Japan SIG Notes HPC-113, 37–42 (2007)
23. Elmroth, E., Tordsson, J.: A Standards-based Grid Resource Brokering Service Supporting Advance Reservations, Coallocation and Cross-Grid Interoperability. J. of Concurrency and Computation 25(18), 2298–2335 (2009)

24. Toporkov, V., Tselishchev, A., Yemelyanov, D., Bobchenkov, A.: Composite Scheduling Strategies in Distributed Computing with Non-dedicated Resources. Procedia Computer Science 9, 176–185 (2012)
25. Moab Adaptive Computing Suite, http://www.adaptivecomputing.com/products/moab-adaptive-computing-suite.php
26. Lee, S.B., Schwartzman, Y., Hardy, J., Snavely, A.: Are User Runtime Estimates Inherently Inaccurate? In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2004. LNCS, vol. 3277, pp. 253–263. Springer, Heidelberg (2005)
27. Toporkov, V., Toporkova, A., Tselishchev, A., Yemelyanov, D.: Slot Selection Algorithms in Distributed Computing with Non-dedicated and Heterogeneous Resources. In: Malyshkin, V. (ed.) PaCT 2013. LNCS, vol. 7979, pp. 120–134. Springer, Heidelberg (2013)
28. Azzedin, F., Maheswaran, M., Arnason, N.: A Synchronous Co-allocation Mechanism for Grid Computing Systems. Cluster Computing 7, 39–49 (2004)
29. Castillo, C., Rouskas, G.N., Harfoush, K.: Resource Co-allocation for Large-scale Distributed Environments. In: 18th ACM International Symposium on High Performance Distributed Compuing, pp. 137–150. ACM, New York (2009)
30. Takefusa, A., Nakada, H., Kudoh, T., Tanaka, Y.: An Advance Reservation-Based Co-allocation Algorithm for Distributed Computers and Network Bandwidth on QoS-Guaranteed Grids. In: Frachtenberg, E., Schwiegelshohn, U. (eds.) JSSPP 2010. LNCS, vol. 6253, pp. 16–34. Springer, Heidelberg (2010)
31. Blanco, H., Guirado, F., Lérida, J.L., Albornoz, V.M.: MIP Model Scheduling for Multi-Clusters. In: Caragiannis, I., et al. (eds.) Euro-Par Workshops 2012. LNCS, vol. 7640, pp. 196–206. Springer, Heidelberg (2013)
32. Moise, D., Moise, I., Pop, F., Cristea, V.: Resource CoAllocation for Scheduling Tasks with Dependencies, in Grid. In: The Second International Workshop on High Performance in Grid Middleware (HiPerGRID 2008), Bucharest, Romania, pp. 41–48. IEEE Romania (2008)
33. Olteanu, A., Pop, F., Dobre, C., Cristea, V.: A Dynamic Rescheduling Algorithm for Resource Management in Large Scale Dependable Distributed Systems. Computers and Mathematics with Applications 63(9), 1409–1423 (2012)

Behavior of Web Servers in Stress Tests

Tomasz Walkowiak

Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-320 Wroclaw
tomasz.walkowiak@pwr.wroc.pl

Abstract. The paper presents an approach to modelling performance and availability of web systems. Based on the set of tests focused on the overutilization of real systems, a three queue model of web server was developed. The model explains the phenomena observed during stress tests. It could be used for simulation based analysis of web systems dependability as well as play a fundamental role in designing an analytical model of web systems under heavy load. The analysis of the developed model resulted in a set of recommendations for stress tests.

Keywords: web server model, performance, availability, stress test.

1 Introduction

Websites become nowadays a part of professional and private lives of almost everybody. As modern websites are complex systems with a wide range of features and access to many external services, meeting expectations of nowadays user is a challenging task for websites providers. One of the most significant problems that websites providers have to face is how they can provide the QoS (quality-of-service) required by their clients, having in mind that the web traffic is highly dynamic and volatile [6]. A given web server configuration may meet requirements under some traffic load while under other workload its performance may not be satisfying. Many aspects like hardware features, software characteristic, and multilayer architecture and network aspects make the problem of analyzing, projecting and reconfiguration of the web systems not a trivial one. Web applications developers to provide users all required functionality are typically using multi-tier architecture, consisting of a web server (Apache, Microsoft IIS, Nginx), an application server (Apache Tomcat, Sun Java System Application Server, IBM WebSphere, JBOSS) and a database server (Oracle, Microsoft SQL Server, IBM DB2, MySQL, PostgreSQL).

Therefore, it is important to model the behavior of web servers since they play the role of the entry point to web applications, especially to model web server performance and its availability. This is the aim of this paper. This problem is studied for many years [7, 10, 11]. Hover proposed solutions agree with real web system behavior for small or medium input load [12]. Within this paper we want to focus on the system behavior under stress tests, i.e. in the range when a system starts to be unstable, dropping requests.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_45,

The paper is organized as follows. We start with an overview of modelling input load of web systems. Then a set of tests of a real web server and noticed unexpected behaviors are presented. It is followed by a three queue model of a web server that explains the observed behavior in the area of server overutilization. Next, the versions of the model for the most common web servers (Apache, IIS and Nginx) are given. Finally, the short summary and recommendations for stress tests are presented.

2 Client Models for Performance Analysis of Web Systems

Within the paper we focus on performance of web systems seen from the user's perspective. It has been proven [9] that if a user will not receive an answer from the website in less than 10 seconds he/she will probably resign from active interaction with the site and will be distracted by other ones. Therefore, a user is not properly serviced if there are too long responses of the web server. This may be caused by a wide range of software bugs, hardware problems, malicious user activities or overload of a website [13].

In this work, performance will be considered in relation to a given load. The question is if the server can handle all incoming requests and, if the response delay is acceptable for the user. The main goal is creation of the model which will allow us to predict the response time of the web service on given load as well as its availability. Availability is understood as the number of properly handled requests (n_{OK}) over all the requests (n) [2]:

$$A = \frac{n_{OK}}{n} \quad . \tag{1}$$

The concept of load tests is generally well known and also implemented in many commercial and open source software tools. One should mention such tools as: Mercury Interactive Load Runner, nowadays called HP LaodRunner, Funkload, Apache JMeter, Rational Performance Tester and Developer Tools from Microsoft.

It is important to remember that a client-server interaction depends a lot on how the traffic is generated by the client. The simplest approach is adopted by the software used for server/website benchmarking. In this case, the server is bombarded with a stream of requests, reflecting the statistics of the software usage. It corresponds to open queuing networks in queue models [5]. Therefore, in most cases the interarrival time is model by the exponential distribution.

Also other types of clients are used in stress tests that correspond to close queuing networks in queue models [5]. In that case the workload is characterized by the number of concurrent clients, sending requests to the server. Each client sends a requests to the server, then it waits for the server to respond (waits for a correct or reject response) and after sends a new request again. The number of clients is kept constants.

The third the most realistic approach is based on modelling of real user behavior and a use of the model within tests. There are a large number of such approaches like User Community Modeling Language [1], Stochastic Formcharts [3] or Realistic Usage Model [14]. But they are out of scope of this paper since we want to describe (model) how the web server behaves in case of stress tests and complex model of clients would add additional phenomena to tests results.

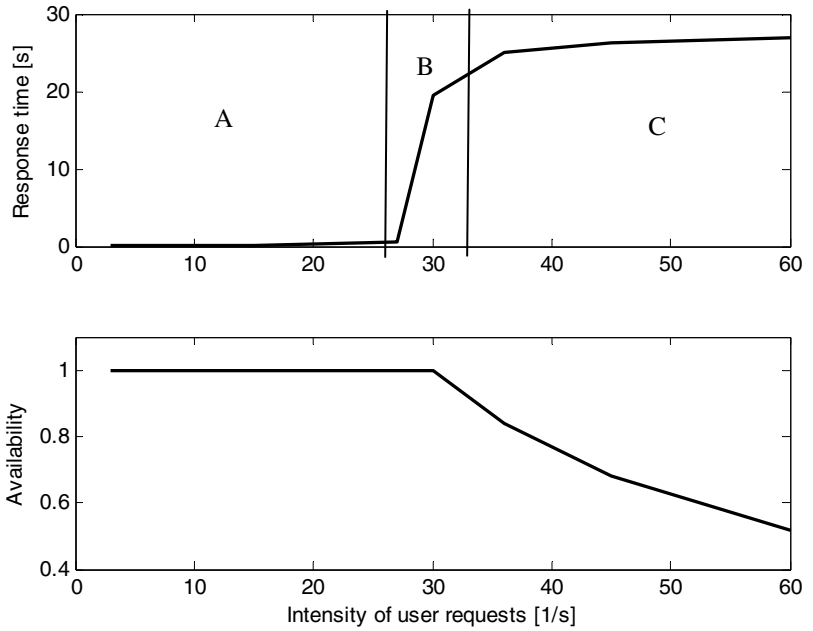


Fig. 1. The performance (the response time and the availability of example web site) under varying intensities of user requests

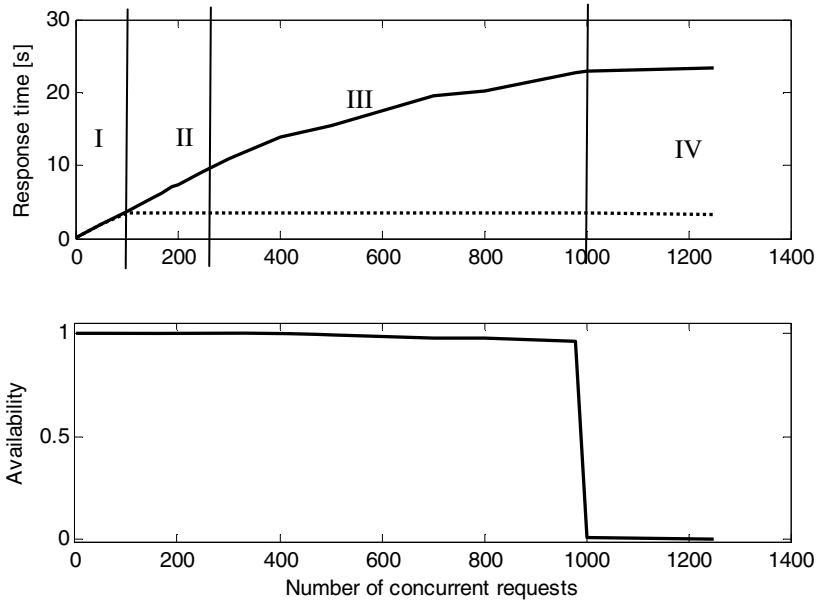


Fig. 2. The performance of an apache server under varying number of concurrent requests

3 Behaviour of Web Servers

3.1 Stress Tests

To model the web server behavior we have considered a simple interaction in a real system. For this purpose, we have set up a simple testbed, consisting of a virtual machine running an Apache server. The server hosts a PHP script application, on which we can accurately regulate the processing time needed to produce a result. This application is exposed to a stream of requests, generated by a client applications (a Python script written by the authors). Full control is maintained of the available processor resources (via the virtualization hypervisor).

At first, test with constant request intensity were performed. The important factor in this approach is the lack of any feedback between the intensity of requests and the server response times. In other words, the client does not wait for the server response, but proceeds to send further requests even if a response is delayed or missing. Fig. 1 shows the results of stress tests. It should be noted that the system is characterized by three distinct ranges. Within the first range, marked as A in Fig. 1, the response time very slowly increases with the intensity of requests. This is the range, where the server processing is not fully utilized: the processor is mainly idle and handles requests immediately on arrival. There is a gradual increase in the response time due to the increased probability of requests overlapping (due to randomness of interarrival time).

In the second range, marked as B, the processing power is fully used up (it could be noticed by monitoring tools), the requests are queued and processed concurrently. The increase in the response time is very fast. It is caused by approaching the processing limit of the system and overloading of web server queues. This is a narrow area, until the server reaches the third phase: overutilization. Within this range (marked as C in Fig. 1) the server is no longer capable of handling all incoming requests. In consequence, some requests are timed-out and some are rejected. Further increase in the request rate does not increase the number of concurrently handled ones. And the response time remains almost constant. On the other hand, the percentage of requests handled incorrectly increases proportionally to the request intensity as it is illustrated in Fig. 1 on the availability plot.

The second type of tests with varying number of concurrent clients (waiting for service response before issuing another request) where performed. Within the tests the server script answers with a time of processing on the server side (time of a PHP script execution). It allows understanding how long a request waits in a queue before it is processed. And how long it is executed on a processor. Results are presented in Fig 2. The upper plot shows the response time, where the solid line shows the total response and the dashed line the server script execution time. The lower plot shows the availability. We could distinguish four ranges in the plot. In the range I, till the moment when the dashed line, i.e. the PHP script processing time, approaches its limit (equal to MaxClients parameter of the Apache server), requests are executed in separate threads. Next, in the range II, the processing time increases in a linear way and the availability is equal to one. The end of this range is equal to MaxClients parameter plus 125. Next, in the range marked by III, the processing time increases, but the

increase slope is not always the same as for range I and II (in could be noticed in Fig. 4). Moreover, within the range III small part of input requests are rejected due to time-out errors. The error response takes usually 189 s or 21 s depending on the version of operating system of the client (regardless the operating system of the server). To understand the origin of such behavior one has to analyze Fig 3. It shows the raw results of tests for 500 concurrent clients. Each point in the plot represents one request. The x-value is equal to a request start time. The whole test duration was set 600 seconds (that's why it is a maximum value on the x axe). The y-value is equal to the response time of this request. It could be noticed that response timea are grouped within a discrete time ranges that differ: 3, 9, 21,… seconds. It is an effect of establishing a TCP/IP connection. It is known as the TCP exponential back-off mechanism, introduced by Jacobson 25 years ago [4] and analyzed in details in many papers, for example in [8].

The IV range presented in Fig. 2 is characterized by almost no further increase in the response time. However, as it could be noticed in Fig 2, the availability drops suddenly to zero. It is caused by the fact that all requests above 1000 are dropped by the server. Moreover, the response time of error requests is very short (several milliseconds) compared to correctly answered requests (several seconds). The used client model results in a fast increase of requests since the client keeps constant number of concurrent requests. So the numerator of (1) is constant whereas denominator grows very fast. To overcome this problem we propose a slight redefinition of definition (1) as:

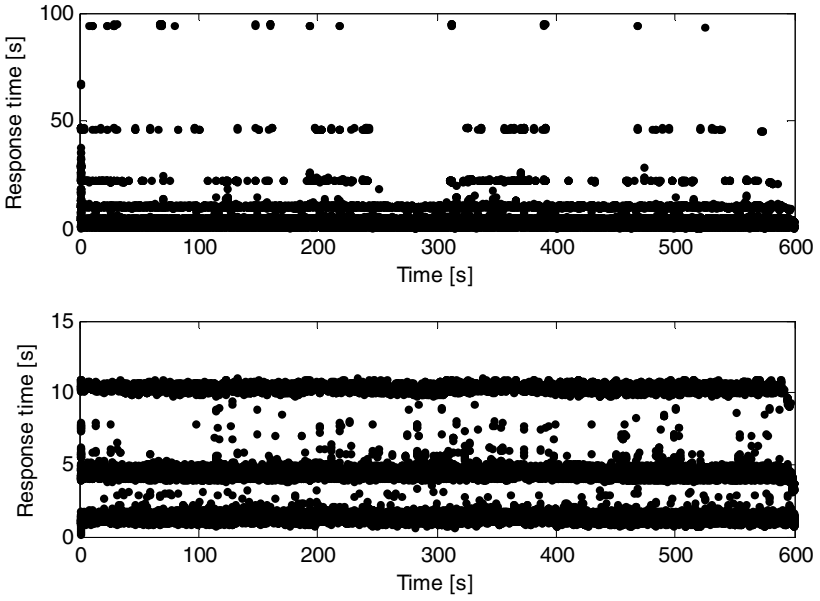


Fig. 3. The response time for an apache server with PHP for 500 concurrent clients (the lower plot is a magnitude of upper plot for the response time in the range to 15 s)

$$A = \frac{1}{C}\int_0^C \frac{n_{OK}(t)}{n(t)}dt \tag{2}$$

whereas $n(t)$ – is a number of requests in the system at time t, and $n_{OK}(t)$ – number of requests in the system that will be correctly responded, C – the length of analysis period. For constant request intensity the definition (2) gives the same results as (1).

3.2 Relation of Performance Results for Different Client Models

The performance analysis for constant request intensities (Fig. 1) and for constant number of concurrent requests (Fig. 2) was done for the same web system. So there is a relation between ranges defined in Fig. 1 and Fig. 2. The range A from Fig. 1 is almost unnoticeable in case of constant number of concurrent requests since in the most part of range A the average number of concurrent requests is less than one. At the end of the range A, the randomness of interarrival time causes that the number of concurrent requests from time to time is more than 1. So the range A is approaching the beginning of the range I in Fig. 2. The range B in Fig. 1 corresponds to ranges I, II and III of Fig. 2. In the range B a web server archives its maximum capability of performance, the whole processor is used. The range C corresponds to range IV from Fig. 2. The differences in the availability plot between ranges C and IV were explained in the previous section as a result of availability definition.

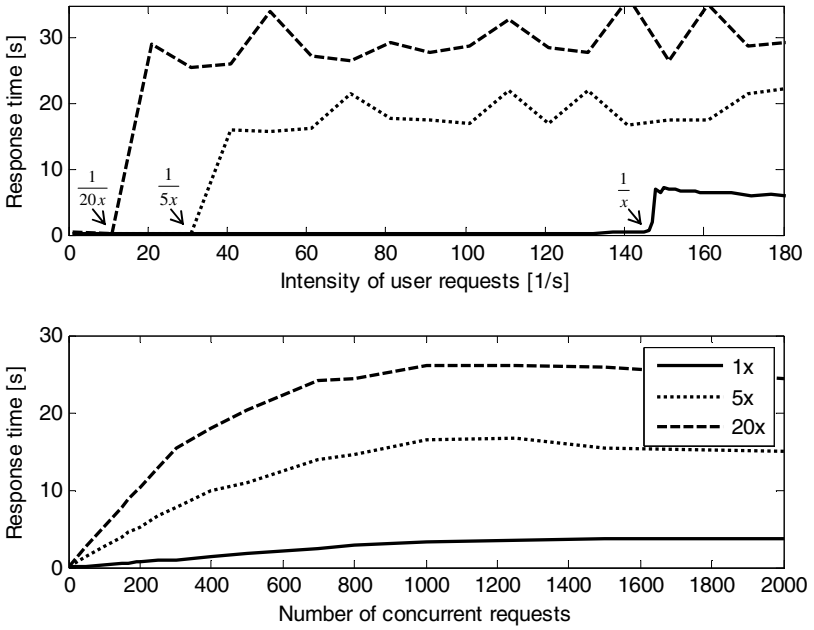


Fig. 4. The performance of an apache server with PHP script under varying task computational complexity

The similarities of response times and availabilities and the simplicity of achieved curves raise the question how to model the performance results. Let's describe the complexity of task performed on a server as a result of the request by the time required to process a single request. Let's mark it by x. To avoid problems with scripts caching it is estimated by calculating the average response time for 10 concurrent requests and divided by 10. The results for different computational complexities of a task are presented in Fig 4. It could be noticed that the overutilization threshold, i.e. the value of intensity when the overutilization starts, is equal to $1/x$ (the upper plot of Fig. 4). Moreover, the slope of the response time in ranges I and II depends proportionally on x (the lower plot of Fig. 4). However, it is hard to estimate the response time in ranges IV and C. Its value depends on x, but the effect of TCP exponential back-off in also noticeable, and especially the timeout limit on the client side (the maximal value of possible timeouts: 21 or 189 s).

Moreover, additional influence of the TCP exponential back-off mechanism could be noticed for relatively small values of x. The phenomena is illustrated in Fig. 5. One can notice that the response time in the range C is dropping down to the value equal to that from the range IV. It is caused by changing proportion of properly answered and rejected requests and delays due to the exponential back-off mechanism. Therefore, there is a need to model the exponential back-off mechanism in details.

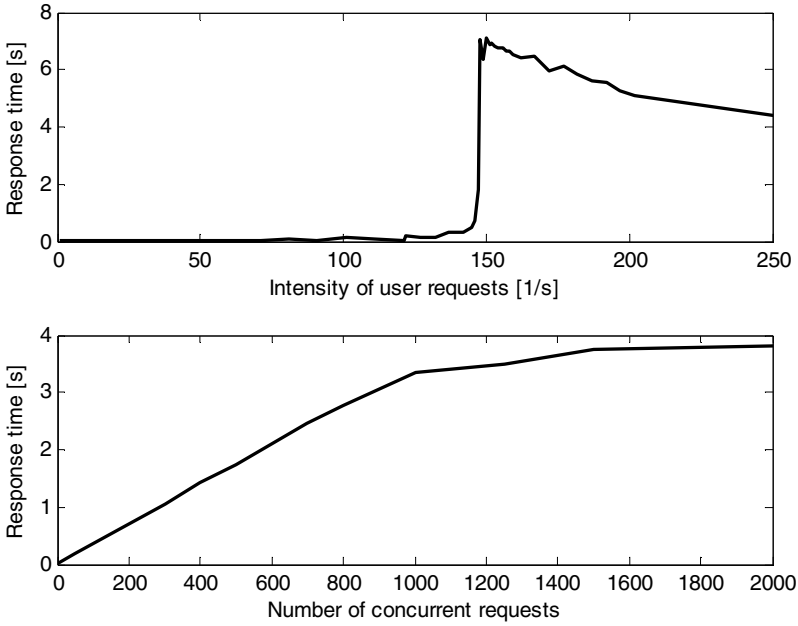


Fig. 5. The performance of an apache server with PHP script under varying number of concurrent clients (the same plot as solid line in Fig. 4)

3.3 Basic Model of the Server Behavior

The analysis of the behavior of an Apache server presented in the previous section led to the formulation of a server behavior model that consists of four elements: the retransmission buffer, the waiting queue, the circular buffer and processor/processors.

The retransmission buffer models the process of establishing TCP connection by a client if a server is not responding. The retransmission buffer works as follows:

1. If the number of processed requests on a server is larger than a given threshold $N_{\max}$ the request is rejected within a few milliseconds (model by a random value).
2. Client for a given time period (Δt) checks if the waiting queue is able to accept a request, if not then goes to step 3, if yes the request goes to the waiting queue.
3. The timeouts parameters are updated:

$$\begin{aligned} \Delta t &= 3 \cdot \Delta t \\ t_d &= 2 \cdot t_d + 3s \end{aligned} \quad (3)$$

4. The client is paused for t_d seconds.
5. If the time elapsed from the beginning of the request processing is longer than the client timeout the request is rejected, if not the processing goes to step 2.

The initial values of timeouts are as follows: $\Delta t = 0.0125\text{s}$, $t_d = 0\text{s}$.

The waiting queue models requests waiting for an execution inside server, it works according to FIFO method and has only one parameter, its length.

Handling of request is done by executing a given task or tasks defined by a request. It is done in time sharing manner and modelled by the circular buffer. In reality concurrent execution is achieved by switching the processors between different tasks. In general it works as follows:

1. If the circular buffer is not full the request is removed from the end of the waiting queue and moved to the circular buffer and execution of a task defined by a request starts.
2. Each task from the circular buffer has an access to a processor (from the set of available one) for a quant of time.
3. The task is finished (therefore removed from the time sharing buffer) when the sum of time quants is larger than the execution time parameter of a given request.

3.4 Modifications of the Basic Model

The above model was based on the most popular web sever: Apache. We have also analyzed other two popular web servers: Microsoft IIS and Nginx. Their behavior could be described by a simplified version of the Apache server model. In case of IIS the response time for constant numbers of concurrent requests enlarges in a linear way till some threshold and all requests above this threshold are rejected immediately [13].

Therefore, there are no retransmission buffer and no circular buffer. In case of Nginx, there is no circular buffer since only one request is executed in parallel, but the retransmission buffer and the waiting queue exists.

3.5 Interaction with Other Services

The operation of all the web based applications is based on the interaction between services. Therefore, it is important to model as well process requests that require calls to other web system components (other web servers of databases). In case of servers that follow the basic model (for example Apache, Tomcat), external calls have an influence on the circular buffer. When a task is waiting for an answer from another server (the request thread is in wait state), the place in the circular buffer is used, but the processor is not. Such behavior results in a situation that the whole circular buffer is used, so new requests are waiting in waiting queue whereas the server is consuming almost no processor power. In case of the modified model (without circular buffer) like IIS or Nginx, the requests waiting for a response from some external server are not using the processor. Therefore, new requests from the waiting queue can be processed. When the response from the external server arrives, the task is placed in an additional FIFO queue. Therefore, the model for web server without circular buffer requires two FIFO queues. It seems that requests are processed from the two queues alternately, but this fact requires further investigation.

4 Conclusions

We have proposed a model of a web server. It describes three most popular nowadays web servers: Apache, IIS and Nginx and explains the servers behavior phenomena noticed during stress tests. The proposed model could be solved by computer simulation using the Monte-Carlo approach and be used to predict the results of any changes in the system configuration [2] on its performance and availability. After including more realistic user model ([1], [3], [14]), it should allow to predict the system performance in a function of session intensities. Moreover, it could play an important role for creation an analytical model (like for example described in [12]) of web systems that will model the web system under a heavy load.

Based on the analysis of the described model we could propose following recommendation for stress tests:

- The client operation system type must be taken into consideration, since it influences the limit of TCP timeouts (21 or 189 s), therefore influences the achieved response time,
- The test for a given number of concurrent clients or a given intensity of requests should last for several minutes to include long timeouts,
- The test results calculation should not include the start and end part of results; it is noticeable in Fig. 3, where the start and end results randomness differ from that in the middle range,
- For concurrent clients the availability should be calculated according to (2),

- For constant intensity of requests tests in the range B should be repeated several times and carefully analyzed, since the server behaves in a chaotic way; the randomness of interarrival times could result in the overloading of the waiting queue so the server starts to use the retransmission buffer (causing long response times) and gets stuck it the overutilization range (range C).

The presented work was funded by the Polish National Science Centre under grant no. N N516 475940.

References

1. Barber, S.: User Community Modeling Language (UCMLTM) v1.1 for Performance Test Workloads. StarEast (2004)
2. Caban, D., Walkowiak, T.: Preserving continuity of services exposed to security incidents. In: Proc. The Sixth International Conference on Emerging Security Information, Systems and Technologies, Rome, pp. 72–78 (2012)
3. Christof, L., Gerald, W.: Modeling a Realistic Workload for Performance Testing. In: Enterprise Distributed Object Computing Conference, Auckland (2008)
4. Jacobson, V.: Congestion avoidance and control. ACM CCR 18, 314–329 (1988)
5. Lavenberg, S.S.: A perspective on queueing models of computer performance. Performance Evaluation 10, 53–76 (1989)
6. Liu, X., Heo, J., Sha, L.: Adaptive Control of Multi-Tiered Web Application Using Queueing Predictor. In: IEEE/IFIP Network Operations and Management Symposium, Vancouver, vol. 5(3), pp. 106–114 (2006)
7. Menascé, D.A.: Load Testing of Web Sites. IEEE Internet Computing 6, 70–74 (2002)
8. Mondal, A., Kuzmanovic, A.: Removing Exponential Backoff from TCP. ACM SIGCOMM Computer Communication Review 38, 19–28 (2008)
9. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
10. Rahmawan, H., Gondokaryono, Y.S.: The simulation of static load balancing algorithms. In: Inter. Conference on Electrical Engineering and Informatics, pp. 640–645 (2009)
11. Slothouber, L.P.: A Model of Web Server Performance. In: Proceedings of the 5th International World Wide Web Conference (1996)
12. Stoczek, W., Walkowiak, T.: An analytical model for 3-tier Web systems performance. In: Information Systems Architecture and Technology: Network Architecture and Applications, pp. 73–82. Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw (2013)
13. Walkowiak, T.: Web server performance and availability model for simulation. Journal of Polish Safety and Reliability Association 3(2), 189–196 (2012)
14. Wang, X., Zhou, B., Li, W.: Model Based Load Testing of Web Applications. In: International Symposium on Parallel and Distributed Processing with Applications, pp. 483–490 (2010)

The Impact of Reconfiguration Time on the Dependability of Complex Web Based Systems

Tomasz Walkowiak and Dariusz Caban

Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-320 Wrocław
{tomasz.walkowiak,dariusz.caban}@pwr.wroc.pl

Abstract. Web based systems are exposed to various dependability threats that affect their availability. The dependability of the systems can be improved by using reconfiguration. We address the problem, how to predict the impact of the reconfiguration time on the availability. For this purpose, semi-Markov processes are utilized. It is demonstrated in a realistic case study that the impact of reconfiguration delays is significant. It has to be considered as a factor when choosing the reconfiguration policy.

Keywords: dependability, web system, reconfiguration.

1 Introduction

Web based systems are quickly becoming critical for the success of all business activities. For this reason, their dependability is crucial. This dependability is severely limited by the occurrence of security incidents, which degrade or compromise the systems operation. There are numerous techniques used to eliminate or reduce the security threats, yet they provide only a limited assurance of faultless access to web services.

Policy-driven reconfiguration is proposed as a technique of extending the system operation time when a security incident occurs [1,3]. This is a well-documented concept, proposed in the NIST Computer security incident handling guide [8] – to "isolate the affected system or network" and to "relocate the target". A practical approach to this technique, utilizing reconfiguration graphs, is analyzed by us in some other publications [2,3].

In this presentation, the impact of the reconfiguration time is considered. Relocating of services is always connected with some period when they are not available to the end-users. This period reduces the overall availability of the services. This can never affect the decision if the system should be reconfigured on failure. It is a factor to consider when determining a reconfiguration strategy.

Due to the random nature of security incidents occurring in web systems stochastic processes are used for mathematical modelling of the system reliability. The most often used stochastic processes in technical system reliability analysis are both Markov and semi-Markov processes [4,5,6]. Markov processes are limited to exponential distributions of all state transitions. This is a reasonable simplification in case of fault

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_46,

occurrences, it is questionable in case of the repair processes, and it is unacceptable in case of the service relocations. To overcome these limitations, the presented analysis is performed using semi-Markov approach.

2 Reconfiguration of Complex Web Based Systems

2.1 Reconfiguration

System configuration is determined by: the hardware (network interconnected hosts that process the software responsible for processing the service components), the communicating service components that are responsible for providing the responses to queries from the end-users and from other service components, and the deployment of the components onto specific hosts. System configuration can be changed by altering the deployment of the service components. A configuration is considered permissible, if the deployment ensures that all the services are performing satisfactorily, i.e. all the requests are handled within their time limits.

Reconfiguration is realized by changing the system from one permissible configuration to another. In normal circumstances reconfiguration is not necessary. On the other hand, it may improve dependability if performed after a fault occurs in the system. In this case, the system is switched to a configuration, in which the fault does not affect its operability.

The faults occurring in the web based systems can have multiple causes. Hardware faults, most commonly considered in dependability analysis, occur relatively rarely due to the advances in manufacturing technology. However, they require long time to repair. The redeployment of services onto computers that are still operational is the temporary remedy to ensure continuity of services.

Most failures in computer systems are caused by exploitation of software faults, which can have consequences ranging from decreased availability to complete blockage of services. Intentional attempts at exploitation of software vulnerabilities constitute the most widespread type of threats to the services[8]. Human mistakes can have effects similar to vulnerability exploitation, even if they are less likely than the intentional attacks.

The normal reaction to all these events is either to restart the affected services or to isolate the affected hosts from the network. The first, if successful, fully recovers the system. The second is in effect equivalent to shutting down all the affected nodes until a remedy can be deployed.

Another serious threat to the system are undirected attacks such as virus or worms proliferation. This type of attack can seriously and unpredictably impact system performance. There are also attacks on services, usually based on draining their limited resources and thus making them unavailable to other users (Denial-of-service attacks [7]). In terms of consequence these attacks are either handled by filtering out the miscreant traffic or are successful, leading to service unavailability.

Effectively, in all these faults the change of configuration may restore the system functionality ensuring service continuity. Notably, this is achieved prior to the recovery from the incident. Of cause, there is a short delay caused by configuration

switching (changing the deployment of services and reconfiguring the system resources to handle the changed service locations). This delay is normally neglected since it is insignificant as compared to the time needed to restore normal system operation. Yet, it may be important when choosing the proper reconfiguration strategy.

2.2 Test Case

Let's consider a fairly simple system to illustrate the proposed approach to dependability analysis. Fig. 1 presents a typical web services system based on four hosts. On top of this hardware infrastructure, there are deployed the various communicating services. Host A is normally running the front-end web service. This service makes use of some web applications located on hosts B and C. The database service, used by the web services, is deployed on host D.

If any of the hosts becomes inoperational (when it is isolated after a security issue), then the affected services are relocated. There are restrictions on this relocation, resulting from the computing and communication resources of the hosts. The front-end service can only be deployed to host A, B or C. Web applications can be relocated between hosts B and C only. The database service can only be relocated to host B. Host B is the only one capable of running all the four services at the same time.

On the basis of the above limitations to permissible reconfigurations, the reconfiguration graph in Fig. 2 is constructed. State *C0* corresponds to all the hosts being operational. States *C1* – *C4* correspond to single hosts being unavailable: A, B, C and D respectively. State *C5* indicates unavailable hosts A and B, state *C6* – unavailable hosts A and C, state *C7* – unavailable hosts C and D, state *C8* – unavailable hosts A and D. State *C9* corresponds to only host B being up.

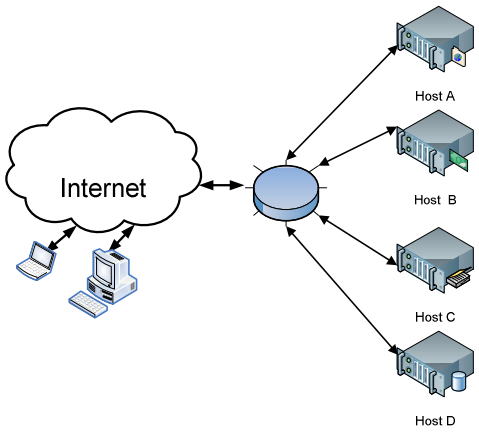


Fig. 1. Test case infrastructure

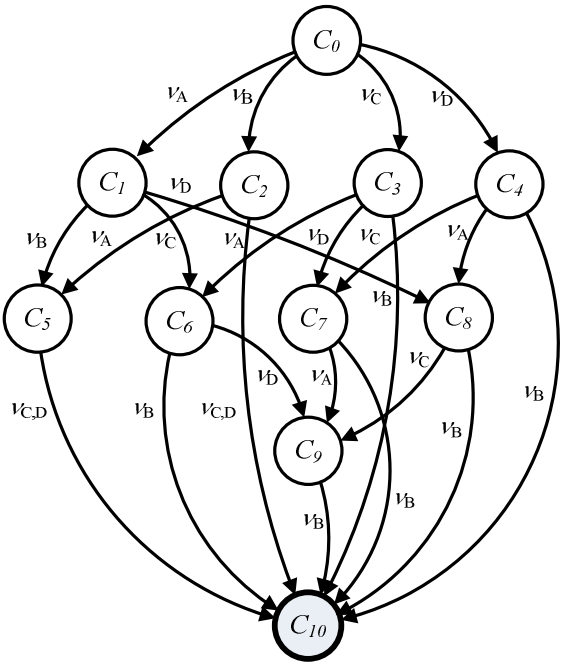


Fig. 2. Reconfiguration graph for the analysed example

3 Dependability Model of Reconfiguration Process

Let's assume the following model of the system behavior: Faults occur in the system randomly, usually with a predictable distribution. Then the system becomes inoperational for some time until reconfiguration procedures restore it to a degraded state. The system is in degraded state until maintenance procedures restore it to full operability. If meanwhile a second fault occurs, then the system again becomes inoperational for some time until the next reconfiguration procedure restores it to another degraded state or the system fails if reconfiguration is not possible. The system is restored from a failure state to full operability after a global repair procedure.

3.1 State Transition Model

The changes occurring in the system can be represented in the form of a state-transition graph. There are 3 categories of system states: operational states corresponding to the various levels of degradation (denoted as S0 – S9 in Fig. 3), a single failure state denoted as S10, and the states when the system is temporarily unavailable during reconfiguration (S11 – S19).

The transitions between the states of the model correspond to the various changes occurring in the system: faults of the hosts, hosts renewal, global renewal after failure, and completion of reconfiguration. Fig. 3 presents an example of a state-transition graph corresponding to the system discussed in 2.2. It should be noted that the graph can easily be derived from the reconfiguration graph. Each transition in the reconfiguration graph has to be split into two transitions with a switching state in between. Then, additional transitions have to be added, corresponding to hosts repair and the global repair. Each transition has to be further described by the distribution of the time of occurrence of the corresponding events (intensity in case of exponential distributions, constant time in case of deterministic transitions).

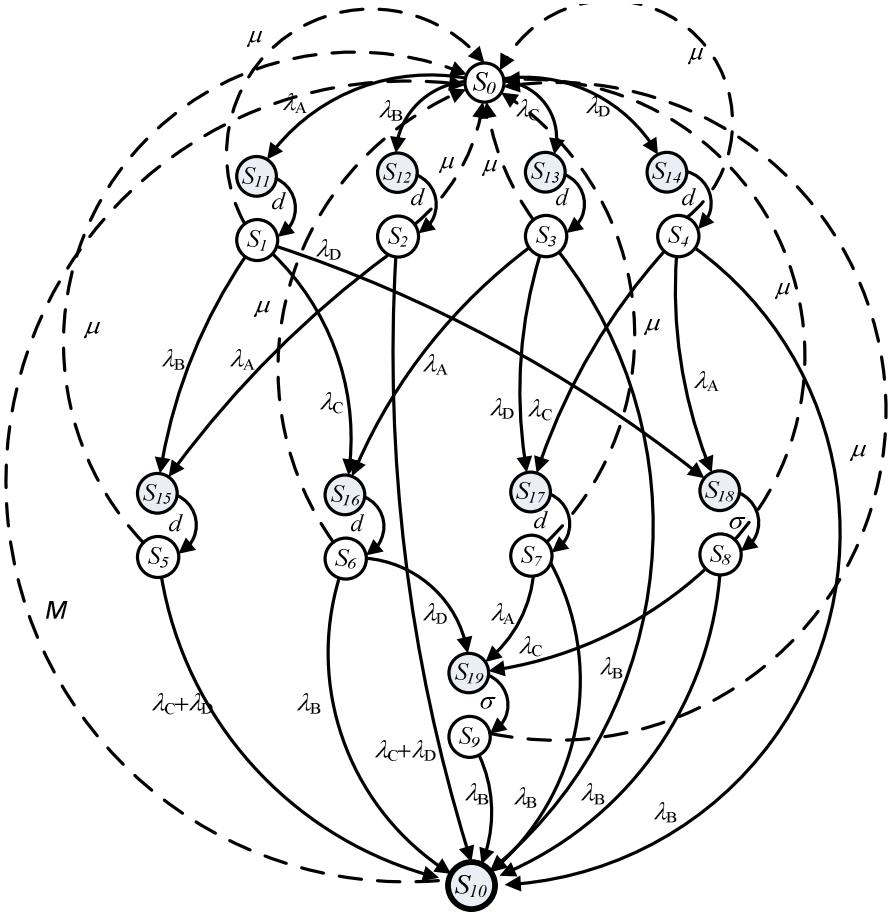


Fig. 3. State transition graph for the analyzed example

3.2 Semi-Markov Analysis

Under the assumptions presented in the previous section a stochastic model describing the process operation of the web system is determined. In case all the transition times either have exponential distributions or they are constant, then the system is described by a semi-Markov process $\{X(t): t \geq 0\}$ with a state space $S = \{0,1,2,..,19\}$. From [5] it follows that in instants of the state changes, the sequence $\{X(\tau_n): n = 0,1,2,...\}$ is a Markov chain with the transition probabilities matrix:

$$\mathbf{P} = [p_{ij}: i, j \in S] \quad , \quad p_{ij} = P\{X(\tau_{n+1}) = j \mid X(\tau_n) = i\}. \tag{1}$$

If T_{ij} represents a duration of state *i* under the condition that the next state will be *j*, the transitional probabilities can be calculated as:

$$p_{ij} = P\left(j = \arg\min_{j':(i,j') \in E} T_{ij}\right). \tag{2}$$

where *E* is a set of ordered pairs of states which represent directed edges of graph in Fig. 3.

According to [6] the limit probabilities for positive recurrent, irreducible, and aperiodic semi-Markov process are given by formula:

$$p_i^* = \frac{\pi_i E(T_i)}{\sum_{i \in S} \pi_i E(T_i)}, \tag{3}$$

where probabilities π_i constitute stationary probabilities of a hidden Markov chain obtained by solving:

$$\forall_{j \in S} \sum_{i \in S} \pi_i p_{i,j} = \pi_j \quad \text{and} \quad \sum_{i \in S} \pi_i = 1, \tag{4}$$

and T_i is a random variable representing duration of state *i*. So,

$$E(T_i) = E\left(\min_{j:(i,j) \in E} T_{ij}\right). \tag{5}$$

The above equation defines the method to calculate the limit probabilities of semi-Markov process states. First of all, the vector of average duration of states $\overline{\mathbf{T}} = \{ E(T_i) \}$ and transition probabilities matrix **P** has to be calculated according to (2) and (5). Next, the set of linear equations defined by (3) and (4) has to be solved.

4 Test Case Analysis

4.1 Exponential Renewals after Reconfiguration

At first, let us analyse a case when the renewal process is described by the exponential distributions. According to the discussion presented in the previous section, the semi-Markov analysis requires the definition of transfer conditional probabilities and expected values of state durations. The calculation of the transfer conditional probabilities has to take into account the structure of the transition graph (Fig. 3). One can notice three different cases in transfers:

i. if there is no transfer from state *i* to *j* then $p_{ij} = 0$;

ii. if there is only one edge starting from state *i* (ending in state *j*) then $p_{ij} = 1$;

iii. if there are more than one edge starting from state *i* then:

$$p_{ij} = P\left(j = \underset{j':(i,j')\in E}{\arg\min} T_{ij} \right) = \frac{\lambda_j}{\sum\limits_{j':(i,j')\in E} \lambda_{j'}}. \tag{6}$$

The formula (6) is based on two facts: (a) the minimum of independent exponentials is exponential and (b) the probability that one exponential (lets marked its intensity by λ_1) is smaller than the other (with intensity λ_2) is equal to $\lambda_1/(\lambda_1+\lambda_2)$.

To simplify the algebraic solution, the intensities of the faults occurrence are assumed equal to λ for faults. Applying these rules to the analyzed test case we achieve:

$$\mathbf{P} = \begin{bmatrix}
0 & 0\,0\,0\,0\,0\,0\,0\,0\,0 & 0 & \frac{1}{4}\,\frac{1}{4}\,\frac{1}{4}\,\frac{1}{4} & 0\,0\,0\,0\,0 \\
1-3\alpha & 0\,0\,0\,0\,0\,0\,0\,0\,0 & 0 & 0\,0\,0\,0 & \alpha\,\alpha\,0\,\alpha\,0 \\
1-3\alpha & 0\,0\,0\,0\,0\,0\,0\,0\,0 & 2\alpha & 0\,0\,0\,0 & \alpha\,0\,0\,0\,0 \\
1-3\alpha & 0\,0\,0\,0\,0\,0\,0\,0\,0 & \alpha & 0\,0\,0\,0 & 0\,\alpha\,\alpha\,0\,0 \\
1-3\alpha & 0\,0\,0\,0\,0\,0\,0\,0\,0 & \alpha & 0\,0\,0\,0 & 0\,0\,\alpha\,\alpha\,0 \\
1-2\beta & 0\,0\,0\,0\,0\,0\,0\,0\,0 & 2\beta & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
1-2\beta & 0\,0\,0\,0\,0\,0\,0\,0\,0 & \beta & 0\,0\,0\,0 & 0\,0\,0\,0\,\beta \\
1-2\beta & 0\,0\,0\,0\,0\,0\,0\,0\,0 & \beta & 0\,0\,0\,0 & 0\,0\,0\,0\,\beta \\
1-2\beta & 0\,0\,0\,0\,0\,0\,0\,0\,0 & \beta & 0\,0\,0\,0 & 0\,0\,0\,0\,\beta \\
1-\gamma & 0\,0\,0\,0\,0\,0\,0\,0\,0 & \gamma & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
1 & 0\,0\,0\,0\,0\,0\,0\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 1\,0\,0\,0\,0\,0\,0\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,1\,0\,0\,0\,0\,0\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,0\,1\,0\,0\,0\,0\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,0\,0\,1\,0\,0\,0\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,0\,0\,0\,1\,0\,0\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,0\,0\,0\,0\,1\,0\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,0\,0\,0\,0\,0\,1\,0\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,0\,0\,0\,0\,0\,0\,1\,0 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0 \\
0 & 0\,0\,0\,0\,0\,0\,0\,0\,1 & 0 & 0\,0\,0\,0 & 0\,0\,0\,0\,0
\end{bmatrix} \tag{7}$$

where: $\alpha = \frac{\lambda}{\mu+3\lambda}$, $\beta = \frac{\lambda}{\mu+2\lambda}$, $\gamma = \frac{\lambda}{\mu+\lambda}$.

The ¼ values in the first row of (7) are caused by applying rule (6) to all four edges starting in state S0 (see Fig. 3) with the same values of intensities.

In case of expected values of the *i-th* state duration we have two cases to consider:

i. if there is only one arc from state *i* and it ends in *j* then $E(T_i) = E(T_{ij})$;

ii. if there is more than one edge starting from state *i* and these transfers are governed by exponential distributions with intensities equal to λ_j then

$$E(T_i) = E\left(\min_{j:(i,j)\in E} T_{ij}\right) = \frac{1}{\sum_{j:(i,j)\in E} \lambda_i}, \tag{8}$$

since minimum of independent exponentials is exponential with the intensity equal to the sum of intensities.

Therefore, the vector of expected values of state durations for the system described by transition graph presented in Fig. 3 is:

$$\overline{T} = \left[\frac{1}{4\lambda}, \overline{\alpha}, \overline{\alpha}, \overline{\alpha}, \overline{\alpha}, \overline{\beta}, \overline{\beta}, \overline{\beta}, \overline{\beta}, \overline{\gamma}, M, d, d, d, d, d, d, d\right], \tag{9}$$

where: $\overline{\alpha} = \frac{1}{\mu+3\lambda}$, $\overline{\beta} = \frac{1}{\mu+2\lambda}$, $\overline{\gamma} = \frac{1}{\mu+\lambda}$.

Solving (3) and (4), we get the limit probabilities which allow us to calculate availability as:

$$A = 1 - \sum_{i=10}^{19} p_i^* \tag{10}$$

$$A = \frac{6\lambda^4 d + 6\lambda^3 d\mu + 22\lambda^3 + 31\mu\lambda^2 + 10\mu^2\lambda + \mu^3}{24\lambda^4 M + 46\lambda^4 d + 22\lambda^3 M\mu + 74\lambda^3 d\mu + 28\lambda^3 + 32\lambda^2 d\mu^2 + 4\lambda^2 M\mu^2 + 31\mu\lambda^2 + 4\lambda d\mu^3 + 10\mu^2\lambda + \mu^3} \tag{11}$$

Fig. 4 presents the results of availability analysis computed from (9) for various values of the mean times of local renewals ($1/\mu$) and reconfigurations (d), where the global renewal time (M) is set to $2/\mu$. It should be noted that the time needed to reconfigure the system has a much greater impact on availability than the renewal times. This shows that it is very important to prepare the reconfiguration strategy as part of the contingency planning in case a security breach occurs in the system. Furthermore, it indicates that the reconfiguration time might be an important factor to be considered when optimizing this strategy.

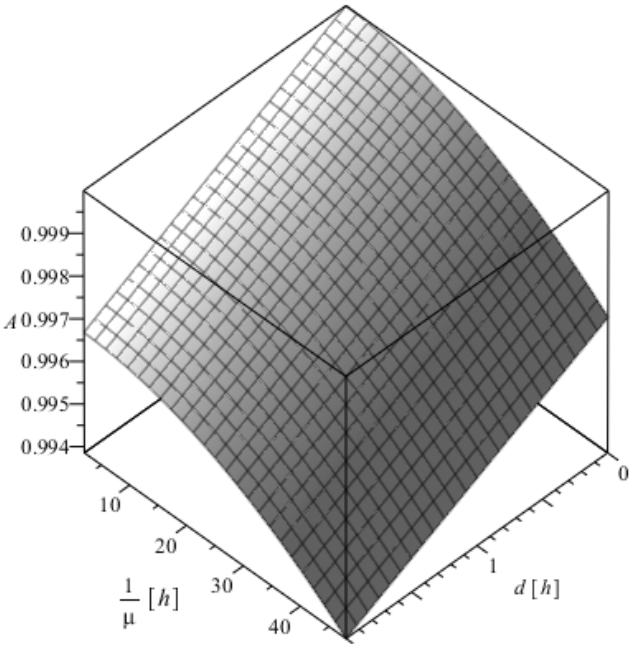


Fig. 4. System availability in the function of the mean renewal ($1/\mu$) and reconfiguration (d) times

4.2 Constant Renewal Times after Reconfiguration

The presented analysis makes it possible to solve a problem with relaxed assumptions regarding the transition distributions. then Markov process assumptions (e.g. the renewal times from state 10 to 1 and the all reconfiguration times might be constant). To show semi-Markov approach capabilities we will change the previous test case by replacing assumption of exponential distribution of reconfiguration renewals (dashed edges in Fig. 3 marked with μ) by a constant value (equal to m).

The calculation of the transfer conditional probabilities requires to add a new case to the analysis presented in the previous chapter: the case when we have constant and exponential transitions (like for state 9 in Fig. 3). Is such cases, the probability that the exponential random value (Λ) is smaller than the constant value is:

$$p_\lambda = P\{\Lambda > \mathrm{M}\} = \int_0^\infty [1 - F_m(x)] dF_\lambda(x) = 1 - e^{-\lambda m}, \tag{12}$$

whereas the transition probability to a state described by the constant value is:

$$p_m = 1 - p_\lambda = e^{-\lambda m}. \tag{13}$$

Taking into consideration the fact that the minimum of independent exponentials is exponential, the transition probabilities, in case of more than one exponential distributions (as for state *1* in Fig. 3) with intensities marked as λ_j, are as follows:

$$p_{\mathrm{M}} = e^{-m\sum_j \lambda_j} . \tag{14}$$

$$p_{\Lambda_k} = \left(1 - e^{-m\sum_j \lambda_j}\right) \frac{\lambda_k}{\sum_j \lambda_j} . \tag{15}$$

The matrix of the transfer conditional probabilities is the same as in the previous considerations (Equation 4), whereas α, β and γ are:

$$\alpha = \frac{1-e^{-3\lambda \mathrm{m}}}{3},\ \beta = \frac{1-e^{-\lambda \mathrm{m}}}{2},\ \gamma = 1 - e^{-\lambda \mathrm{m}} . \tag{16}$$

The average state durations for transitions described by the exponential distribution and constant value can be calculated as:

$$E\left(\min\{\Lambda, \mathrm{M}\}\right) = \frac{1-\mathrm{e}^{-\lambda \mathrm{m}}}{\lambda} . \tag{17}$$

In case of several exponentials we have:

$$E\left(\min\{\Lambda_1, .., \Lambda_j, \mathrm{M}\}\right) = \frac{1-\mathrm{e}^{-\mathrm{m}\sum_k \lambda_k}}{\sum_k \lambda_k} \tag{18}$$

Therefore, the parameters $\bar{\alpha}$, $\bar{\beta}$ and $\bar{\gamma}$ of the expected values of state durations (9) are:

$$\bar{\alpha} = \frac{1-\mathrm{e}^{-3\mathrm{m}\lambda}}{3\lambda},\ \bar{\beta} = \frac{1-\mathrm{e}^{-2\mathrm{m}\lambda}}{2\lambda},\ \bar{\gamma} = \frac{1-\mathrm{e}^{-\mathrm{m}\lambda}}{\lambda} . \tag{19}$$

Solving (3) and (2), we get the limit probabilities which allow us to calculate the availability of the system with reconfiguration. The result is demonstrated, as compared to the values obtained in 4.1, by the graph in Fig. 5. As it can be noticed the difference in system availability calculated for exponential renewals after reconfiguration and constant ones is very small (less than $9*10^{-5}$) for the realistic values of reconfiguration and renewal times.

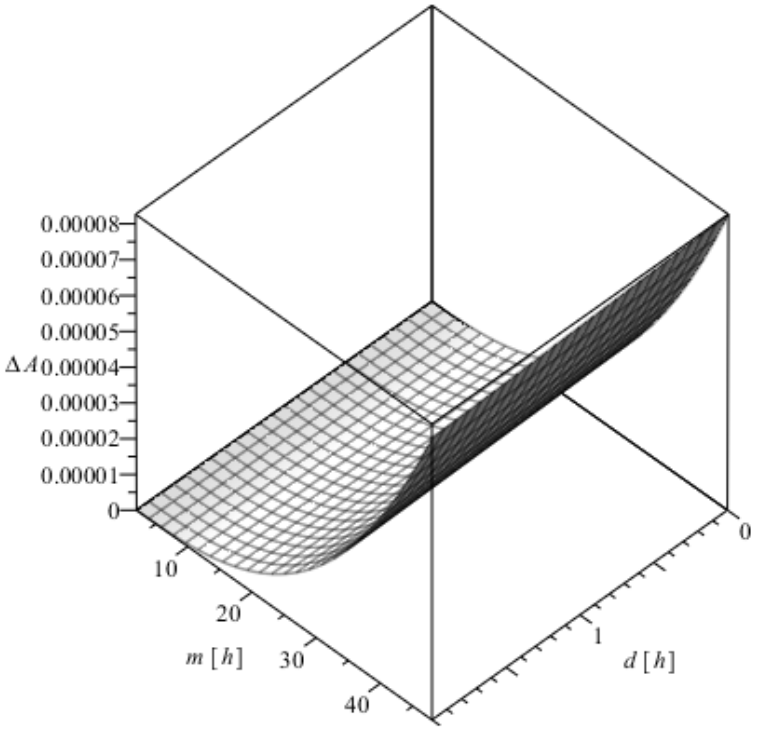


Fig. 5. Comparison of system availability for exponential and constant renewals after reconfiguration, for various mean renewal (m) and reconfiguration (d) times

5 Conclusions

We have proposed a technique for determining the impact of reconfiguration time on availability of web based systems, exposed to breaches of security, especially when those breaches lead to cicatrisation of hosts. The technique is based on semi-Markov process analysis, which allows non-exponential reconfiguration time distributions.

It is demonstrated that the reconfiguration time significantly impacts the overall availability. Fairly short reconfiguration times decrease the availability noticeably, though never to the extent that reconfiguration does not improve dependability.

The introduction of semi-Markov approach was justified by the fact that reconfiguration time is never exponentially distributed. The conducted numerical experiments show, though, that the assumed distributions lead to assessments that differ at the level of measurement accuracy.

Acknowledgement. The presented work was funded by the Polish National Science Centre under grant no. N N516 475940.

References

1. Aime, M.D., Pomi, P.C., Vallini, M.: Policy-driven system configuration for dependability. In: Proc. 1st International Workshop on Dependability and Security in Complex and Critical Information Systems DEPEND 2008, Cap Esterel, pp. 420–425 (2008)
2. Caban, D., Walkowiak, T.: Preserving continuity of services exposed to security incidents. In: Proc. The Sixth International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2012, Rome, pp. 72–78 (2012)
3. Caban, D., Zamojski, W.: Dependability analysis of information systems with hierarchical reconfiguration of services. In: Proc. 1st International Workshop on Dependability and Security in Complex and Critical Information Systems DEPEND 2008, Cap Esterel, pp. 350–355 (2008)
4. Grabski, F., Jaźwiński, J.: Semi-Markov models of safety of the renewal systems operation. Journal of KONBiN 3(6), 153–176 (2008)
5. Grabski, F.: The reliability of the object with semi-Markov failure rate. Applied Mathematics and Computation (2003)
6. Kulkarni, V.G.: Modeling and analysis of stochastic systems. Chapman & Hall, London (1995)
7. Lent, R.: Evaluating a migration-based response to DoS attacks in a system of distributed auctions. Computers & Security 31(3), 327–343 (2012)
8. Scarfone, K., Grance, T., Masone, K.: Computer Security Incident Handling Guide, rev. 1. NIST Special Publication, 800–61 (2008)

Propagation Losses in Urban Areas

Marian Wnuk and Leszek Nowosielski

Military University of Technology, 2 Gen. S. Kaliskiego Str. 00-908 Warsaw Poland
{leszek.nowosielski,marian.wnuk}@wat.edu.pl

Abstract. In the article a modified UTD method for determining propagation losses in the big-city area has been presented. Modification of this method consists in using a model of physical geometry of wave propagation and using patterns occurring in the UTD method for reflection coefficients and diffraction of waves at corners. Calculation of resultant field takes place with the use of the superposition method. On the basis of the conducted analysis calculation algorithm has been suggested and sample calculations have been presented.

Keywords: UTD model, propagation.

1 Introduction

In the first decade of the 21st century one can observe a sudden development of personal cordless communication systems. It causes the increase of demands concerning supply of high quality signals with as little power output as possible. These demands are satisfied with the use of long-time well-known cordless telephony. Development of cordless telephony systems is also directed at increase of possibilities and service improvement. Because of that network operators are more and more interested in solutions concerning wave propagation in a built-up area. There is a large number of models allowing to determine propagation losses, however each of them has its advantages and disadvantages. On the basis of the conducted comparative analysis it was found that the most optimal model for determining propagation losses in urban area can be the UTD (Uniform Theory of Diffraction) model described by Pathak [2] and Crane [1] allowing to take into account all phenomena occurring during propagation of electromagnetic wave in cities.

2 Phenomena Accompanying Propagation

Propagation models are necessary while analyzing propagation conditions in radio communication networks. Therefore those models should reflect significant and characteristic features of propagation conditions in an examined system.

Based on the model one can forecast distribution of electric field in an area of system operation. To forecast distribution of field in urban area one should take into account a large number of extra factors which have a significant influence on propagation of radio waves in this area (Fig. 1). These are the following factors:

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_47,

- Reflection or dispersion of radio waves incident on boundary surfaces of two media and waves propagating in multilayer medium,
- Infiltration (penetration) of waves deep inside an adjacent medium which is accompanied by refraction and attenuation of waves while passing of wave through boundaries of media with various electric parameters,
- Diffraction causing deviation of the course of wave at edges of narrow gaps, shields or on surfaces of solids with curvatures compared to wave length,
- Interferences of two or more waves with identical frequency resulting from occurrence of the above mentioned phenomena.

While analyzing propagation conditions, apart from the above mentioned phenomena, one should also take into account a kind of urban environment which is strongly differential. Generally four categories of urban building development can be distinguished:

- Dense and high development of big city downtowns,
- Relatively low development of medium and little cities as well as suburban development,
- Housing estates with low houses around areas with compact development,
- Area of rural character surrounding built-up areas.

The listed environment categories significantly differ in conditions of radio-wave propagation. Those conditions depend on position of base station antenna and mobile station towards each other and development surrounding them. Wave propagation can take place above roofs of buildings in case of highly lifted base station antenna. If that antenna does not tower over development, radio-waves propagate below roofs among streets.

Analytical methods appropriate in studies of wave propagation, particularly distribution of field radiated by transmitters, can be divided in two basic groups: deterministic methods and empirical methods. The first group includes modern methods of "ray tracing". They are based on approximation of geometrical optics and uniform theory of diffraction (UTD). Obtaining solutions on the basis of analytical dependences used in those theories requires knowledge of: geometry of surroundings and obstacles (buildings and their components, structural components) occurring in the area of wave propagation and values of their electrical parameters in an analyzed frequency band.

3 Modified UDT Propagation Model in Urban Environment

UDT propagation model is a three-dimensional propagation model for microcellular radio communication in urban environment. This model takes into account multiple reflections of the type: wall-wall, wall-ground, ground-wall and diffraction at building corners as well as multiple reflections of diffracted signals. It is based on uniform theory of diffraction. Ray geometry is quite complex due to occurrence of reflections from ground surface and a number of reflections and diffractions on walls, building edges and ground surface. A basic difficulty in forming this three-dimensional model is determining an accurate point of reflection on surface and an accurate point of

diffraction on edge and appropriate incidence surfaces. It is necessary to calculate polarization components of reflected and diffracted rays and their further trajectories. In each point of reflection or diffraction a system of coordinates with "determined ray" or "determined edge" has been used as well as double matrices of reflection or diffraction coefficients. Good conformity of theoretical results with practical measurements indicate correctness of the UTD method in modelling propagation in urban radio communication. The UTD model is often used in practical applications. For example it was used in applications described by Bajda [3], Byłak [4], Laskowski [5] and Byłak [6].

A sample geometry of propagation model has been shown in Fig. 1 where T_x means location of transmitter on the main street of width W_1, and side street with width W_2 is located within the distance d_1 from point T_x.

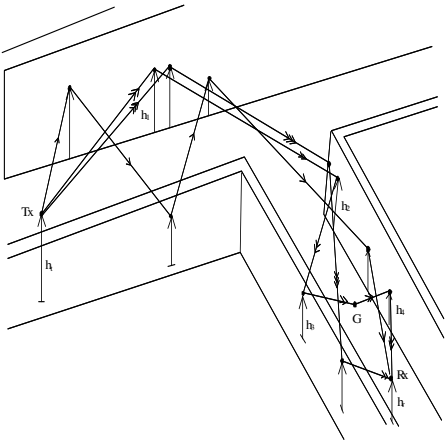


Fig. 1. Geometry of wave propagation

By the streets there are high buildings which walls, as it was assumed, are smooth surfaces with mean permittivity ε and conductivity σ. Transmitter is placed at height h_t over the street and within the distance x_o from the right wall.

Vertically polarized receiving antenna R_x is placed at height h_r over the street surface and its position is displaced along the main street (visibility area LOS) to the side street (optical invisibility area OOS). In Fig. 1 three typical ray paths for wave reaching R_x in invisibility area are shown.

We assume that in an accepted coordinate system the center of system overlaps with the place of transmitter position T_r, a x, y & z are described in the following way:

- x determines boundary points localized at street edge,
- y determines direction along which propagation takes place.

Proceeding to mathematical recording, wave equation describing propagation of electromagnetic wave in such a street canyon can be written in the following form:

$$\left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right)E = f(x,y,z,t) \tag{1}$$

where domains of particular coordinates are described in the following way:

$$\begin{aligned} x &\in \left(-\left(w - x_p\right), x_p\right) \\ y &\in \left(-\infty, +\infty\right) \\ z &\in \left(0, z - h\right) \end{aligned} \tag{2}$$

Whereas function is defined in the following form:

$$f(x,y,t) = \delta(x)\delta(y)\phi(t)\delta(z-h) \tag{3}$$

We set up the following initial and boundary conditions:

$$\begin{aligned} E &= \partial tE = 0 \\ E\left(x_p, t\right) &= 0 \\ E\left(-\left(w - x_p\right), t\right) &= 0 \end{aligned} \tag{4}$$

Using properties of entire function, definition of Laplace transform and Fourier transform the above wave equation can be written in the following form:

$$E\left(x_p, \bullet\right) = 0$$

$$E\left(-\left(w - x_p\right)\right) = \left\{ \begin{array}{l} Ae^{\gamma x_p} + Be^{-\gamma x_p} = -ce^{-\gamma\left|x_p\right|} \\ Ae^{-\gamma\left(w - x_p\right)} + Be^{\gamma\left(-w - x_p\right)} = -ce^{-\gamma\left|-\left(w - x_p\right)\right|} \end{array} \right\} \tag{5}$$

We receive solution of this system of equations with the use of Kramer formulas and after change of variables:

$$\overline{E}(x,y,z,s) = \frac{1}{4\pi^2}\iint \frac{\Phi(s)}{2\gamma}\left[\sum_{n=0}^{\infty} e^{-\gamma x n} e^{i(\alpha y + \beta(z-h))}\right] d\alpha d\beta = \frac{1}{8\pi^2}\sum_{n=0}^{\infty}\sum_{k=1}^{3}\left[\int_{-\infty}^{\infty}\Phi(s)dq\int_{-\infty}^{\infty}\frac{e^{-\gamma x_{nk} + i\omega r}}{\gamma}d\omega\right] \tag{6}$$

In order to calculate the above mentioned integrals, we use Cauchy-Goursate theorem, Jordan's lemma and then we carry out integration along real axis ω on contour Γ, which is chosen in such a way to fulfil the following equality:

$$\gamma x_{nk} - i\omega r = s\tau \quad \text{for} \quad \tau \rangle 0\,, \tag{7}$$

where γ- means poles of integrands included within contour Γ.
It results from the above that the searched solution of wave equation is the below formula:

$$E(x,y,z,t) = \frac{1}{4\pi}\sum_{n=0}^{N}\sum_{k=1}^{5} H\left(t - \frac{R}{c}\right)\frac{1}{R}\Phi\left(t - \frac{R}{c}\right) \tag{8}$$

4 Solution for the Case of Propagation of Streets Crossing at an Angle Different from Right Angle

An interesting for analysis scenario seems to be propagation of electromagnetic wave in street canyon, where receiver is situated in the street diverging at some angle

(different from 90°) from the place of transmission T_x. The above scenario is shown in Fig. 2. In the Figure the following markings have been used:

- P_d – diffraction lower point,
- P_g – diffraction upper point,
- Φ_p – incidence angle of wave ray formed as a result of diffraction in point P_d,
- Φ_p' – reflection angle of wave ray formed as a result of diffraction in point P_d,
- Φ_l – incidence angle of wave ray formed as a result of diffraction in point P_g,
- Φ_l' – reflection angle of wave ray formed as a result of diffraction in point P_g,
- γ_l – azimuth angle,
- α_0 – corner angle.

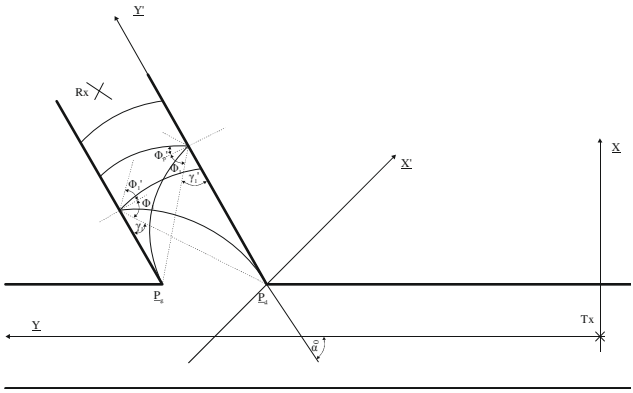


Fig. 2. Wave propagation along street canyon

Solution of wave equation in real coordinates is the function:

$$E(\rho_{n,p,l},t)=e^{i\omega t}\left[\sum_{n=0}^{Np}\sum_{p=0}^{1}(-1)^p\frac{C_{0np}}{4\pi\rho_{n;p}}e^{-ik\rho_{n;p}}+\sum_{n=0}^{Nl}\sum_{l=0}^{1}(-1)^{l+1}\frac{C_{0nl}}{4\pi\rho_{n;l}}e^{-ik\rho_{n;l}}+\frac{C_0}{4\pi\rho}e^{-ik\rho}\right], \quad (9)$$

where:

$$\rho=\sqrt{x^2+y^2+(z-h_{Tx})^2}$$
$$\rho_{np;l}=\sqrt{(x_{np;l})^2+y^2+(z-h_{Tx})^2} \quad (10)$$
$$x_{np}=(-1)^p x+2(n+1)W_1-2p(W_1-x_p) \qquad p=0,1$$
$$x_{nl}=(-1)^l x+2(n+1)W_1-2lx_p \qquad l=0,1$$

Indices p mean reflections of waves from the right wall, whereas l mean reflections of waves from the left wall. Amplitudes of waves reflected in the street of width W_1 take values:

$$C_{onp}=C_0\prod_{i=0}^{n}R_{ip}A_i(s) \qquad C_{onl}=C_0\prod_{i=0}^{n}R_{il}A_i(s) \quad (11)$$

Analogically for diffraction waves in the street of width W_2 we receive:

$$E(\rho x) = e^{i\omega t}\left[\frac{C_0^{(1)}}{4\pi\rho_1}e^{-ik\rho_1} + \frac{C_0^{(2)}}{4\pi\rho_2}e^{-ik\rho_2} + \sum_{n=0}^{Np}\sum_{p=0}^{3}\frac{C_{np}^{p}(-1)^{p}}{4\pi\rho_{np}}e^{-ik\rho_{np}} + \sum_{n=0}^{Nl}\sum_{l=0}^{3}\frac{C_{1n}(-1)^{l+1}}{4\pi\rho_{1n}}e^{-ik\rho_{1n}}\right] \quad (12)$$

On the basis of geometric dependences binding geometric dimensions of streets and height of antenna installations and their mutual location we can determine radius-vectors $\rho_1 \cdots \rho_{3n}$.

Amplitudes of diffraction waves in the street of width W_2 take values:

$$C_{np} = C_0^1 \prod_{i=0}^{n} D_{ip}\overline{A_i}(s) \quad C_{1n} = C_0^1 \prod_{i=0}^{n} D_{il}\overline{A_i}(s) \quad (13)$$

For the above dependences coefficient of expansion for reflection from surface $A_i(s)$ can be calculated from the below dependence.

$$A_i(s) = \frac{\rho}{(s+\rho)} \quad (14)$$

Coefficient of expansion for diffraction from surface $\overline{A_i}(s)$ can be calculated from the below dependence.

$$\overline{A_i}(s) = \sqrt{\frac{\bar{\rho}}{(\bar{s}+\bar{\rho})}} \quad (15)$$

In order to determine angles $\varphi_p, \varphi_l, \varphi^*{}_p, \varphi^*{}_l$ of reflected waves and angles $\overline{\varphi_p}, \overline{\varphi_l}, \overline{\varphi_p}^*, \overline{\varphi_l}^*$ for diffraction waves we use equations describing fronts of particular incident waves and reflected from the street walls.

In case of waves propagating with the street of width W_1 general equation of wave front takes the form:

$$\rho_{n,p;l} = ct \quad (16)$$

$$\sqrt{(x_{n,p;l})^2 + y^2 + (z - h_{Tx})^2} = ct \quad (17)$$

where:

– c velocity of electromagnetic wave.

Vector normal for wave front is described by dependence:

$$\vec{N}_{n,p;l} = \left[\frac{\partial\rho_{n,p;l}}{\partial x}, \frac{\partial\rho_{n,p;l}}{\partial y}, \frac{\partial\rho_{n,p;l}}{\partial z}\right] \quad (18)$$

In overt form the above dependence is determined by formulas:

– for the right wall (indices p)

$$\vec{N}_{n,p} = \left[\frac{(-1)^p x_{np}}{\rho_{np}}, \frac{y}{\rho_{np}}, \frac{z - h_{Tx}}{\rho_{np}}\right] \quad (19)$$

– for the left wall (indices l)

$$\vec{N}_{n,l} = \left[\frac{(-1)^l x_{nl}}{\rho_{nl}}, \frac{y}{\rho_{nl}}, \frac{z - h_{Tx}}{\rho_{nl}} \right] \tag{20}$$

As it can be easily noticed, both vectors $\vec{N}_{n,p}$ and $\vec{N}_{n,l}$ are unit vectors i.e. $\left|\vec{N}_{n,p}\right| = 1$ and $\left|\vec{N}_{n,l}\right| = 1$, so their coordinates (projection on axes OX, OY, OZ) express direction cosines of angles which are formed by those vectors with appropriate axes. This fact can be written with the use of dependence:

$$\vec{N}_{n,p,l} = [\cos\alpha, \cos\beta, \cos\gamma] \tag{21}$$

where:

– α angle which is formed by vector $\vec{N}_{n,p,l}$ with axis OX,
– β angle which is formed by vector $\vec{N}_{n,p,l}$ with axis OY,
– γ angle which is formed by vector $\vec{N}_{n,p,l}$ with axis OZ.

In plane $Z = h_{Tx}$ it can be presented as in Fig. 3.

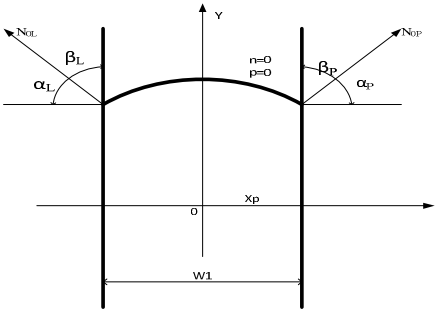


Fig. 3. Wave propagation along street canyon

The above figure shows that vector normal to wall is parallel to axis OX, whereas proper ray corresponds with direction of normal vector.

The method of determining angles of wave propagating within the street of width W_1 completely moves to determining reflection geometry of waves propagating in the street of width W_2, while coordinates X, Y, Z should be replaced by $\overline{X}, \overline{Y}, \overline{Z}$ expressed with the use of matrices:

$$\begin{bmatrix} \overline{x} \\ \overline{y} \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \cos\alpha_0 & -\sin\alpha_0 & 0 \\ \sin\alpha_0 & \cos\alpha_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{22}$$

The result obtained below is an initial basis for considering the phenomenon of diffraction occurring on an object of any shape. A shape of an object closest to the shape

of building (rectangle or square) was taken for analysis. Electrical distribution is determined by the below dependence.

$$E(x,y,z,t)=\frac{1}{4\pi}\sum_{n=0}^{N}\sum_{k=1}^{5}H\left(t-\frac{R_{nk}}{c}\right)\frac{1}{R_{nk}}\Theta\left(t-\frac{R_{nk}}{c}\right) \tag{23}$$

5 Study and Analysis of Algorithm for Determining the Signal Level Value

Below an algorithm for determining signal values in the place of reception has been presented. Its action bases on a modified UTD method. An application determining signal values was implemented in MATLAB environment. The application was written for propagation environment presented in the mentioned below figure.

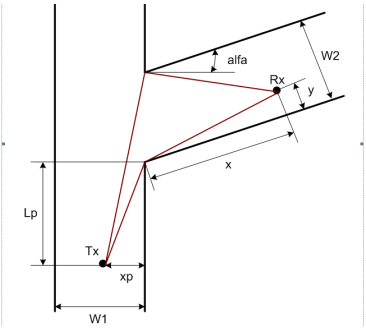


Fig. 4. Geometry of propagation environment used in the program

Fig. 4 presents an intersection of two streets forming canyon in which signal propagates while reflecting from its walls and bending at its edges. In the street of width W_1 transmitter T_x is located and receiver R_x in the street of width W_2. Signal path has been marked in red colour (reflections from buildings in the street are not marked). In the Fig. 4 also some parameters are visible which can be changed for appropriate matching of environment to real conditions.

6 Presentation of Computer Simulation and Measurement Results

Diagrams showing the results of program operation are presented below. It is a set of diagrams illustrating basic rules of signal propagation and signal values in reception points for each from the mentioned above environment. Moreover diagrams comparing results obtained with the use of computer simulation and measured in real environment are shown.

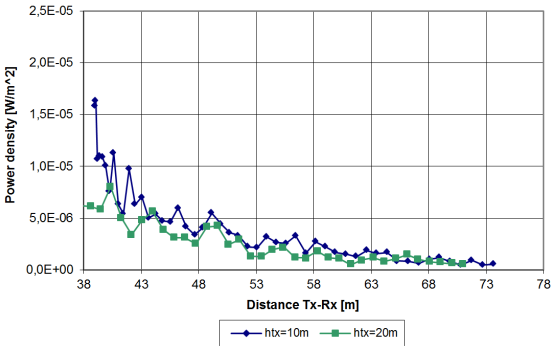


Fig. 5. Dependence of signal value on height of transmitter (simulation results)

The above diagram presents dependence of signal value on distance between transmitter and receiver for intersection of streets, for two values of transmitter height. As it can be seen for the same distances T_x-R_x signal for h_{tx}=20m is weaker. It is caused by the fact that in an established environment between T_x and R_x there are no obstacles. Therefore transmitter placed higher does not "facilitate" signal to reach the point of reception. However at the same time the higher placed transmitter causes extension of propagation path and thus decrease of signal value.

In order to obtain a more reliable analysis of program operations, comparisons of theoretical results returned by the program with real measurements have been made. Those measurements were made in the centre of the big city by one of cellular network operators. Propagation environment in which comparison of results was made is similar to surroundings presented in Fig. 4. Analysing the obtained diagrams it is clear that theoretical calculations show lower values of signal power in comparison with actual data. It might be caused by several conditionings such as: influence of nearby transmitters (working in the same/different system), accuracy of environment mapping.

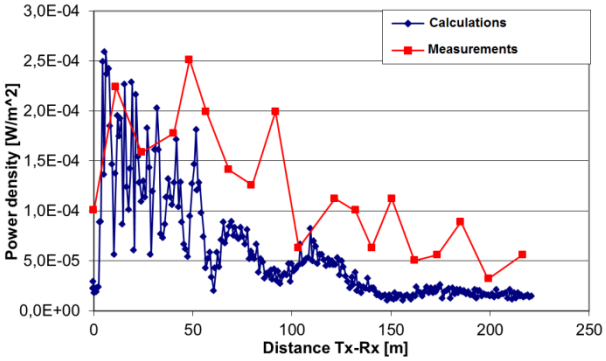


Fig. 6. Comparison of actual losses and theoretical calculations

7 Conclusions

Modelling propagation losses in urban areas is difficult and requires a suitable factual preparation. In those areas the most visible is influence of many propagation phenomena, which can significantly change a broadcast signal. Therefore their appropriate categorization and taking them into account during studies is very important.

Radio signals propagate in accordance with four mechanisms: reflection, dispersion, refraction and diffraction. The result of the above mentioned phenomena is possibility of description of radio communication system with the use of three, almost independent of each other, phenomena: changes of losses of signal power in the function of distance, slow masking and quick fading connected with propagation multipath.

In this study emphasis has been put on the UTD method. It is widely used in microcells, in which knowledge of environment geometry allows to increase efficiency of this method significantly. Moreover it is simple in implementation into a computer program. In that way a tool can be created due to which obtaining results which make designing cellular systems easier will be very simple and quick.

References

1. Crane, R.K.: Propagation Handbook for Wireless Communication System Design. CRC Press (2003)
2. Pathak, P.H., Kouyoumjian, R.G.: A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface. Procedings of the IEEE 62(11) (November 1974)
3. Bajda, A., Wrażeń, M., Laskowski, D.: Diagnostics the quality of data transfer in the management of crisis situation. Przeglad Elektrotechniczny 87(9A), 72–78 (2011) ISSN 0033-2097
4. Byłak, M., Laskowski, D.: Assessment of network coding mechanism for the network protocol stack 802.15.4/6LoWPAN. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) New Results in Dependability & Comput. Syst. AISC, vol. 224, pp. 75–82. Springer, Heidelberg (2013)
5. Laskowski, D., Łubkowski, P., Kwaśniewski, M.: Identification of suitability services for wireless networks. Przeglad Elektrotechniczny 89(9), 128–132 (2013) ISSN 0033-2097
6. Byłak, M., Laskowski, D.: Diagnosis coding efficiency of network coding mechanism for wireless networks. Przeglad Elektrotechniczny 89(9), 133–138 (2013) ISSN 0033-2097
7. Piotrowski, Z.: Angle phase drift correction method effectiveness. In: Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, pp. 82–86 (2009) ISBN: 978-83-62065-00-4
8. Bugaj, M., Przesmycki, R., Wnuk, M., Piwowarczyk, K.: Analysis of methods measuring attenuation of RF line. Przeglad Elektrotechniczny 88(2), 17–19 (2012)

Web Service for Data Extraction from Semi-structured Data Sources

Marina V. Yashina and Ivan I. Nakonechnyy

Moscow Technical University of Communication and Informatics,
8-a Aviamotornaya str., Moscow, Russia
{yash-marina,ivan.nakonechnyy}@yandex.ru

Abstract. Data extraction methods are relevant because of the development of network technologies, particularly the Internet. From the viewpoint of data transmission and processing, these processes generated difficulties during designing subsystems interactions due to the difference in the representation of the original data format. The paper is devoted to improve the data extraction and classification of semi-structured data presented in spreadsheet form. The main problem of that process is that in public sources that kind of data is often presented in a human-readable form, which is completely different from a classical normal table forms. The paper explores the problem of capturing semi-structured data in Information System SSSR-Ed in the setting of education on the example of processing schedules in a spreadsheet. Algorithms and system solutions to this problem are proposed.

1 Introduction

At the present time the problem of data extraction from semi-structured data sources becomes much more significant. Internet has become easiest and often the only source of information for ordinary people. At the same time an abundance of information is accompanied by its quite chaotic representation. Usually web-developers don't care about possibility of centralized provision of information to user, as a result, user have to view many sites to get the right information.

Semi-structured data is a form of structured data that does not conform to the formal structure of data models associated with relational databases or other forms of data tables, but nonetheless contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data[1].

The paper is devoted to improvement of the data extraction and classification of semi-structured data presented in spreadsheet form. The main problem of this process is that in public sources that kind of data is often presented in a human-readable form, which is completely different from a classical normal table forms. Data extraction is the act or process of retrieving data out of unstructured or semi-structured data sources. It is important to understand the difference between the concepts of data extraction and data mining. The purpose of data extraction is the collecting and systematization of data, while data mining is the extracting knowledge from data.

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_48,

Therefore, data extraction sometimes reduced to converting human readable data in machine readable.

The project REC "Theoretical Problems of intellectual systems creation for monitoring and control of distributed processes"[7] under the leadership of Buslaev AP client-server system SSSR was developed.

System SSSR (Smartphone, Server, Student, Distribution) is a technology for automated collection of information through mobile devices, smartphones and tablets, sending multimedia information to the server in a structured form. Theoretical framework and client-server systems for distributed monitoring was developed, and later SSSR-Education system for monitoring learning processes in higher education were developed as a result of work.

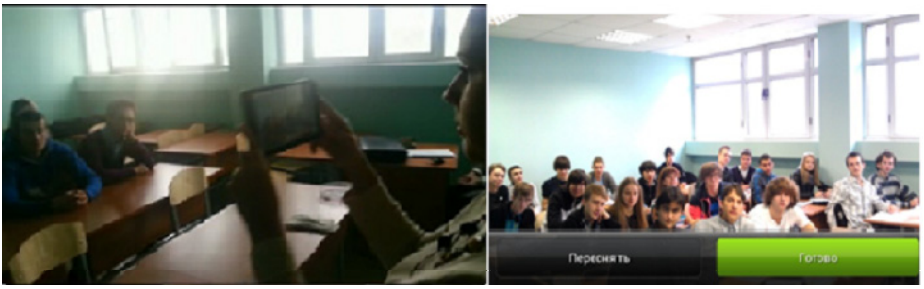


Fig. 1. SSSR-Ed Operator and students

The main objectives of SSSR-Ed are to monitor the learning process in real time, its synchronization with the schedule, the introduction of statistics for visits and student performance. Smartphone communicating with the server gets online mode, information on employment in accordance with the actual time. Therefore, you cannot enter the relevant data, either before or after class.

The main advantage is the lack of binding to a specific operator's working environment for the implementation of employee engagement and base. As yet it was decided to use a variety of mobile devices running operating systems Android and iOs. Thus, multimedia data can be transferred from any student, even if universities are territorially remote. Requirements for client devices are modest, enabling even low cost smartphones and tablets. Android devices require an operating system 2.1 and above. iOs devices require version 4 and above. The system suffers from the need for a permanent connection to the Internet.

SSSR-Ed system provides support of the educational process. Teachers use a client application on a smartphone, which is connected to the server.

Application represents the following information:

- The name of the class;
- Duration of the class;
- Group’s students;
- Theme of the class.

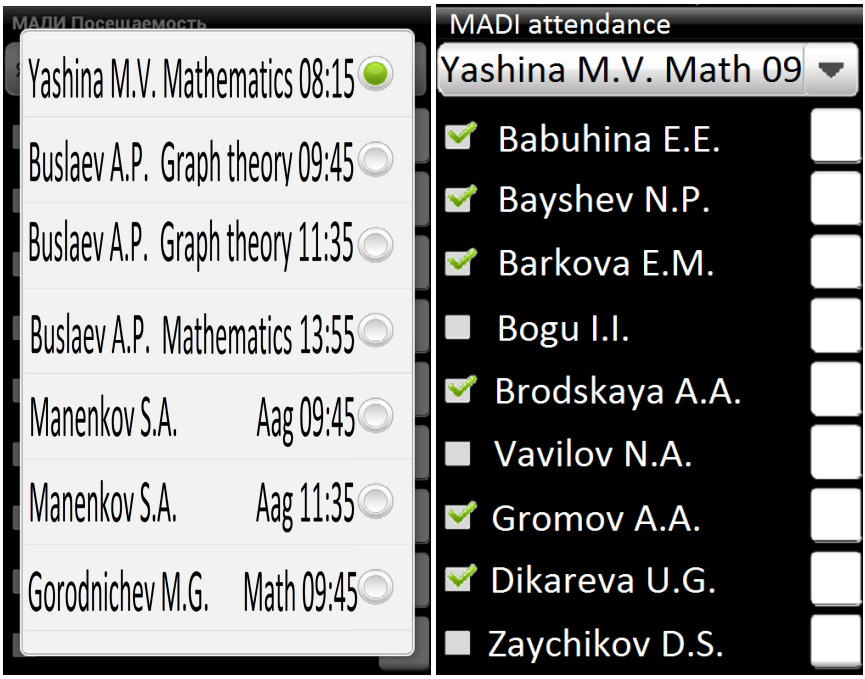


Fig. 2. User Interface

The advantage is in that server communicates with the participants during the production process, creates a new process with the occurrence of certain events (teacher's disease, schedule's changing, decline in academic performance/student attendance, etc.). Teacher may also request additional information on teaching materials, tasks, etc. If a teacher is sick, replacing it has full access to stories and lessons and plans. This organization provides increased reliability of the educational process.

All these processes require periodic updates at the beginning of the semester or day. We were faced with the task of organizing the automatic data refresh schedule. In the process of solving this problem we had to solve the following problems:

- A table is not in any of the normal forms.
- The problem of the presence of additional information. The presence of additional commentary and information that does not requiring treatment.
- The problem of coherence cells in meaning. The presence of associations between cells in explicit and implicit form, i.e. violation of the relational model.
- The problem of determining the content type. The lack of strict format of the text inside the cell.
- The problem of determining the relationship between logical entities.
- The problem of interoperability between systems. Need for automated import.
- The problem of data updating. Because data source may change required to continuously monitor sources.
- Conversion Issue "human-readable data" in machine-readable.

Our solution can be represented as follows:

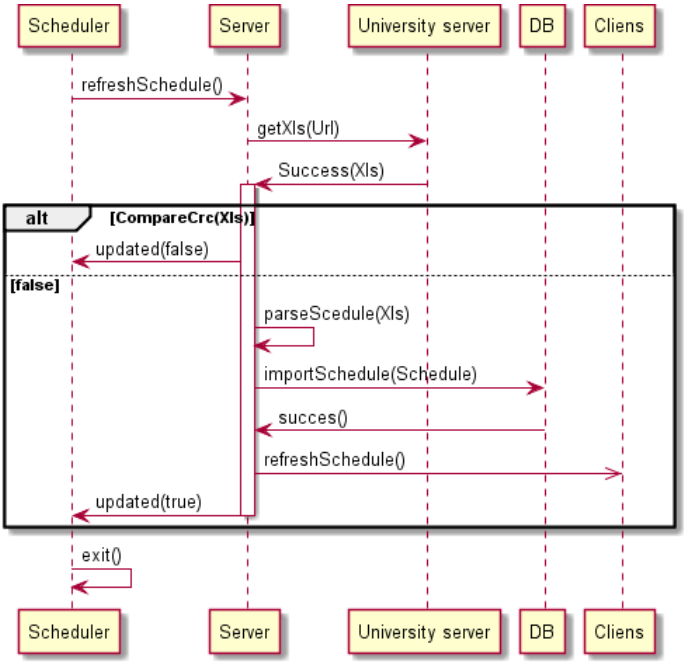


Fig. 3. Schedule updating UML sequence diagram

Actions sequence:

1. Application server periodically takes the schedule data from the server of the University;
2. Checksum is compared with the data obtained previously.
3. If the data matches, the process is completed, if not, the data is parsed with visual, text and semantic analysis. More about these operations is see in section 4 of this article.
4. The result is compatible with the XML, provides the schedule object model.
5. This structure is loaded into the database to the relevant tables.

In this article we faced the problem of analyzing the data matrix (schedule spreadsheet) provided by the dean's office as XLS-file for automatically import into SSSR-Ed system. The main difficulty was that, the schedule was presented and readable to humans, but the automatic insertion is not possible for a number of problems:

In [8], [9] presented systems for road services. In this paper we considering the questions of information systems in the field of education, and algorithms for extracting useful information from semi-structured source, for example, automatic extraction of schedule of classes from a spreadsheet into our system of support of the educational process “SSSR-Ed” [10],[11].

2 Data Extraction Model

2.1 Data Extraction Problems

Retrieving related data is consistent solution of four problems:

- Data extraction problem
- Structure synthesis problem
- Data mapping problem
- Data integration problem

Despite the fact that in this article we consider the extraction of data from a spreadsheet document, these four steps are applicable to data mining in general. For example, when solving the problem of extracting data from web sites have to alternately pass the same four points.

If you open any page of schedule, you see something like the following picture:

«Confirm»
President MTUCI
Prof. Adgemov A.S. _____
«____»_____ 2012 r.

SCHEDULE
1 semestr 2012/13
Faculty TF-2 Course 230100, 090900, 230400 Year 1

GROUPS		GROUP1301	GROUP1302	GROUP1303	GROUP1304	GROUP1305
Monday	9.30-11.05		Foreigh Language cr. 216			PHYSICAL CULTURE
	11.15-12.50	AaG lect. Manenkov S.A. cr. 301				Math. an. sem. cr. 308
	13.00-14.35	PHYSICAL CULTURE		AaG sem. cr. 224	Math. an. sem. cr. 517	PPP sem. cr. 223 Informatics sem. cr. 511
	15.00-16.30	History sem. cr. 316			Math. an. cr. 526 lect. Andreeva N.P. PPP sem. cr. 512	
	16.40-18.10					
Tuesday	9.30-11.05		Informatics lab.		RYaKR sem. cr. 219	AaG sem. cr. 314
	11.15-12.50	Informatics sem. cr. 518a	PHYSICAL CULTURE		AaG sem. cr. 522	RYaKR sem. cr. 514
	13.00-14.35	Informatics lect. Kuvikina M.. cr.511				
		RYaKR sem. cr. 508	Foreigh Language sem. cr.401	Informatics sem. cr. 511	AaG lect. Andreeva N.P. cr. 522	
	15.00-16.30	Mathematics lect. Manenkov S.A. cr. 344			PHYSICAL CULTURE	Foreigh Language sem. cr. 216a
	16.40-18.10		Informatics sem. cr. 511			

Fig. 4. Schedule source data

To a user is not difficult to understand relevant data for extraction, but you need to teach this machine. So, in the case of data extraction, recognition problem can lead to the problem of finding duplicate data structures. If we teach the algorithm to find any repetitive data structure, then probably it would be that the person considers "relevant data". Implementation of the algorithm determining repetitive data structures - is an

additional topic for discussion and all existing (published) algorithms, even in relatively simple have a lot if mistakes.

There are several conceptually different approaches to the implementation of the algorithm for determining the repetitive data structures:

- Based on the text content
- Based on the semantically markup
- Based on the construction of a hash for each of the elements and groups of elements with similar hashes
- Based on the visual image

Clearly, these approaches are rarely used, often used the combined solution. In our work we combine all these methods except hashing algorithms.

For our system, we have developed a system integrating data from third-party sources or provided by the university founded in social networks.

2.2 Algorithm for Finding Related Data

Our main goal is to integrate various types of information provided in the form of spreadsheets, in our system. We have decided to use the so-called "decision trees" for the definition and separation of the objects according to their degree of connectivity. Particular, to convert human-friendly tables to machine - readable does not require any very sophisticated algorithms. The algorithm can be described as consistent execution of four steps:

1. Determination of the source data. Due to additional explanatory data, it is necessary to separate the useful information from the comments.
2. Separation of the individual objects. To a user is not difficult to understand whe the information ends up one object and starts on another spreadsheet using the appropriate styles. This may be the selection of borders, text style, merging cells.
3. Data classification. Once we got a set of individual objects, you must determine whether they are connected to the properly class.
4. Formation of a hierarchical tree. After classification of objects it is possible to form a tree of subordination among themselves. Subsequently, the same data are converted to XML.

2.3 String Matching Algorithms

String-matching is a very important subject in the wider domain of text processing. String-matching algorithms are basic components used in implementations of practical software aggregating under most operating systems. Moreover, they are emphasize programming methods that serve as paradigms in other fields of computer science (system or software design). Finally, they also play an important role in theoretical computer science by providing challenging problems.

Approximate String Matching. In computer science, approximate string matching (often colloquially referred to as fuzzy string searching) is the technique of finding strings that match a pattern approximately (rather than exactly). The problem of approximate string matching is typically divided into two sub-problems: finding approximate substring matches inside a given string and finding dictionary strings that match the pattern approximately.

By directly searching requires no specific occurrences, and one-pattern string matching algorithms and dictionary searching are not suitable, so we needed to use regular expressions.

2.4 Regular Expressions

Once we got the hierarchical tree, we propose to use “regular expressions” recognizers for further useful information. “Regular expressions” - formal language for search and manipulation of the substring in the text, based on the use of metacharacters. In fact, this line-sample consisting of characters and metacharacters and set the rules for your search.

This approach requires prior specific user patterns and pattern recognizers for the data, but can significantly increase the accuracy of the final data.

3 Data Extraction System for SSSR-Ed

3.1 Problem Formulation

The project REC "Theoretical Problems of intellectual systems creation for monitoring and control of distributed processes" under the leadership of Buslaev AP client-server system SSSR was developed.

Fig. 5. SSSR-Ed System

System SSSR (Smartphone, Server, Student, Distribution) is a technology for automated collection of information through mobile devices, smartphones and tablets, sending multimedia information to the server in a structured form. Theoretical framework and client-server systems for distributed monitoring were developed, and later SSSR-Education system for monitoring learning processes in higher education was developed as a result of work.

3.2 Search Method in a Spreadsheet Using SSSR-Ed

The source Data. Initially, we had data in a spreadsheet, below. We have separate the process of solving these problems to the following steps.

Step 1. Unnecessary data problem. The first step occurs trimming table. We solved this problem by finding repeating patterns of data (periodic structures) in the limit of a row or column. Based on analysis of multiple tables, you can say with certainty that the first row and first column, as a rule, are headers for the rest of the table. At this stage it is sufficient to find one repeating row and one duplicate column to get the range of cells containing useful information.

GROUPS		GROUP1301	GROUP1302	GROUP1303	GROUP1304	GROUP1305
Monday	9.30-11.05		Foreigh Language cr. 216			PHYSICAL CULTURE
	11.15-12.50	AaG lect. Manenkov S.A. cr. 301				Math. an. sem. cr. 308
	13.00-14.35	PHYSICAL CULTURE		AaG sem. cr. 224	Math. an. sem. cr. 517	PPP sem. cr. 223 / Informatics sem. cr. 511
	15.00-16.30	History sem. cr. 316			Math. an. cr. 526 lect. Andreeva N.P. / PPP sem. cr. 512	
	16.40-18.10					
Tuesday	9.30-11.05		Informatics lab.		RYaKR sem. cr. 219	AaG sem. cr. 314
	11.15-12.50	Informatics sem. cr. 518a	PHYSICAL CULTURE		AaG sem. cr. 522	RYaKR sem. cr. 514
	13.00-14.35	Informatics lect. Kuvikina M.: cr.511				
		RYaKR sem. cr. 508	Foreigh Language sem. cr.401	Informatics sem. cr. 511	AaG lect. Andreeva N.P. cr. 522	
	15.00-16.30	Mathematics lect. Manenkov S.A. cr. 344			PHYSICAL CULTURE	Foreigh Language sem. cr. 216a
	16.40-18.10		Informatics sem. cr. 511			

Fig. 6. The data set after the first step

Step 2. Cell coherence problem. Display style object definition. In this step, a determination by the logical coherence cells sense. The main idea is that for a person clear separation bins in style (text style, edge detection, etc.) unlike machines. The table has been observed that essentially all the individual lines are allocated in thickness, so the cell was removed style to style an attribute that is responsible for the presence / line width. Thus, it was possible to get proper cell arrays belonging to the same entity.

GROUP1301	GROUP1302	GROUP1303
	Foreigh Language cr. 216	
AaG lect. Manenkov S.A. cr. 301		

Fig. 7. Definition of a connection between the cells

As we can see, for a given data element uses the relationship between the horizontal-governmental means spreadsheet cells, but there is no connection between the vertical, it can be determined only by the characteristic separation.

Step 3. Problem of finding relationships between data. At this stage, as well as on the first, we are looking for duplicate data structures, but in order to find the basic data, which later become column names and key relationships.

		GROUP1301	GROUP1302	GROUP1303
Monday	9.30-11.05		Foreigh Language cr. 216	
	11.15-12.50	AaG lect. Manenkov S.A. cr. 301		
	13.00-14.35	PHYSICAL CULTURE		AaG sem. cr. 224
	15.00-16.30	History sem. cr. 316		
	16.40-18.10			

Fig. 8. Interconnected data

Step 4. The problem of determining the content type. At this stage, were actively used the so-called regular expressions. For example, a teacher can be easily removed from any line following regular expression. [A-Z]{0,1}[a-z]{0,}[ ]{0,1}[A-Z]{0,1}[.]{0,1}[A-Z]{0,1} [.]{0,1}.

For the rest: the type of lessons are always the first word (if there begins with a lowercase letter), if the first word starts with a capital is the subject and the type of lecture classes. Subject of the first word if the lecture begins with a capital letter, or second. Clasroom can be found as A *, cr. *.

For the example shown above data will be decoded as:
Instructor: Manenkov S.A.
Subject: AaG
Groups: Group1301, Group 1302, Group 1303
Class room: 301

```
<Cell>
    <Teacher>Manenkov S.A.</Teacher>
    <ClassRoom>301</ClassRoom>
    <Lesson>AaG</Lesson>
    <LessonType>Lecture</LessonType>
</Cell>
```

Fig. 9. The data obtained in the form of XML

Step 5. The problem of determining the connection between logical entities. After determining the types of cells, we can construct a connectivity graph.

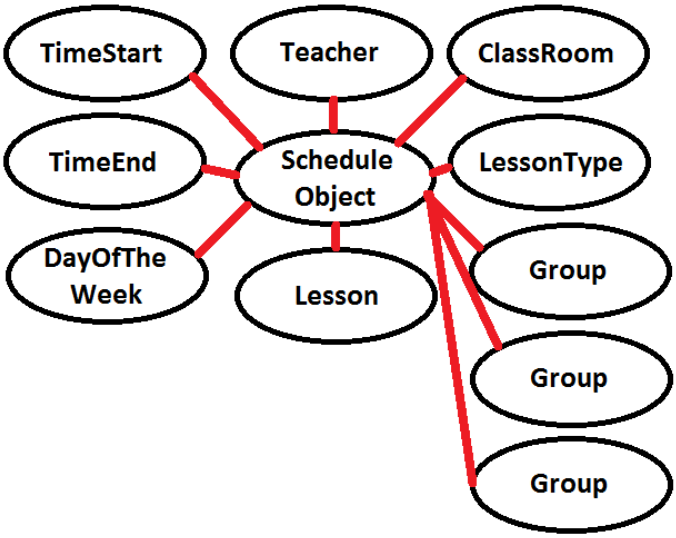


Fig. 10. Connectivity graph

Constructing the adjacency matrix, we can easily determine the entities and relationships between objects.

Step 6. The problem of interaction between systems. After all the steps we obtain a data structure of the following form:

```
<ScheduleElement>
    <DayOfWeek>Monday</DayOfWeek>
    <TimeStart>11:15</TimeStart>
    <TimeEnd>12:50</TimeEnd>
    <Teacher>Manenkov S.A.</Teacher>
    <ClassRoom>301</ClassRoom>
    <Lesson>AaG</Lesson>
    <LessonType>Lecture</LessonType>
    <Groups>
        <Group>Group1301</Group>
        <Group>Group1302</Group>
        <Group>Group1303</Group>
    </Groups>
</ScheduleElement>
```

Fig. 11. The resulting data set

Such a structure is easily processed and exported to external information systems.

Source code. In general, the code can be represented as follows:.

```
HSSFWorkbook workbook = new HSSFWorkbook(file);
HSSFSheet sheet = workbook.getSheetAt(0);
List<List<Cell>> parsedDataTable = parseDataTables(sheet);
List<List<Object>> parsedObjects = new ArrayList<List<Object>>();
        for(List<Cell> cellist : parsedDataTable)
        {
            List<Object> objectsList = new ArrayList<Object>();
            for(Cell cell : cellList)
            {
                objectsList.add(analyzeCell(cell));
            }
            parsedObjects.add(objectsList);
        }
List<List<Object>> linkedObjects = findDependence(parsedObjects);
Schedule schedule = formSchedule(linkedObjects);
Dictionaries.addSchedule(schedule);
```

4 Conclusion and Future Work

In this paper, we investigate the problem of capturing semi-structured data in SSSR-Ed on the example of processing schedules in an spreadsheet. Algorithms and system solutions to this problem are proposed. In that work we combine full and semi-automatic methods for data extraction to ensure greater relevance of the information retrieved. According to test results we obtained the results:

- Definition of objects and their connections in spreadsheets makes sense to relying on the understanding that human-understandable data are not shared by the cells, but their styles.

- Regular expressions are a powerful tool for data retrieval, which does not allow determining the relatedness of objects, but provides its classification.
- To determine the entities and relationships between them is the most convenient tool for matrix connection.

Currently the system is being tested at the Department MKiIT Moscow Technical University of Communication and Informatics and the Department of VM Moscow Automobile and Road Institute. We create a request for a registration of patent of utility model by category: "Method of automated support of the educational process."

Future work. We plan to continue to expand and improve our experience of interaction with external systems that do not have standard interfaces for data exchange. We plan to connect social networking to get more information about the participants in the educational process. We also consider the possibility of connecting digital libraries combined with advanced annotation system by participants of educational process, it will optimize the process of learning the material through search process optimization of related materials from different sources.

Acknowledgments. This work could not be completed without the help and support of Buslaev A.P.

References

1. Buneman, P.: Semistructured Data. In: Symposium on Principles of Database Systems (1997)
2. Arens, Y., Chee, C.Y., Hsu, C.-N., Knoblock, C.A.: Retrieving and integrating data from multiple information sources. International Journal of Intelligent and Cooperative Information Systems
3. Hsu, J.Y.-J., Yih, W.-T.: Template-based information mining from html documents. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence
4. Smith, D., Lopez, M.: Information extracting for semistructured documents. In: Proceedings of the Workshop on Management of Semi-Structunzd Data,
5. Liu, L., Pu, C., Han, W.: An XML-enabled data extraction toolkit for web sources
6. Li, Z., Ng, W.K., Sun, A.: Web data extraction based on structuralsimilarity
7. Bugaev, A.S., Buslaev, A.P., Kozlov, V.V., Yashina, M.V.: Distributed Problems of Monitoring and Modern Approaches to Traffic Modeling. In: 14th International IEEE Conference on Intelligent Transportation Systems (ITSC 2011), Washington, USA, October 5-7 (2011)
8. Buslaev, A.P., Abyshov, R.G., Kupriyanov, U.D., Yashina, M.V.: A distributed system for monitoring road maintenance. Vestnik MADI 24(1), 79–85 (2011)
9. Abyshov, R.G., Buslaev, A.P., Nakonechnyy, I.I., Yashina, M.V.: Registered in the Registry of the computer programs. Patent SSSR-AN (February 22, 2012)
10. Buslaev, A.P., Burikova, T.A., Guseva, A.S., Nakonechnyy, I.I., Yashina, M.V.: Application of information - computer networks for monitoring of complex socio-technical processes for the example of evaluating the quality of maintenance of knowledge in mathematics in high school. Part 2. Technology certification. Handbook, 72 c. Tehpoligraftsentr (2013)
11. Buslaev A.P., Nakonechnyy I.I., Yashina M.V.: Patent claim SSSR-Education № 2013153089 от (November 29, 2013)

Investigation of System Reliability Depending on Some System Components States

Elena Zaitseva, Vitaly Levashenko, and Miroslav Kvassay

Faculty of Management Science and Informatics,
University of Zilina, Univerzitna 8215/1, 01026 Zilina, Slovakia
{Elena.Zaitseva,Vitaly.Leavshenko,
Miroslav.Kvassay}@fri.uniza.sk

Abstract. The system reliability depending on some system components states changes is investigated in this paper. This investigation assumes the representation of the initial system by the structure function. This function definition agrees to the Boolean function. Therefore the mathematical approach of Logical Differential Calculus is used in the analysis of the system reliability change depending on the changes of components states. Based on this mathematical approach, calculation of two measures is considered – Dynamic Reliability Indices and Birbaum's Importance Measure. These measures are indices of importance analysis, that allow estimating the system reliability depending on components states changes.

Keywords: Reliability, binary-state system, logical differential calculus.

1 Introduction

Reliability evaluation methods exploit a variety of tools for system modeling and reliability indices calculation. A discrete model has been used in reliability analysis frequently. There are two mathematical types of this model: a *Binary-State System* (BSS) and a *Multi-State System* (MSS). The system and its components are allowed having only two possible states (completely failed and perfect functioning) in a BSS. This approach is well known in reliability analysis, but can prevent the examination of many situations where the system can have more than two distinct states [1-3]. MSS reliability analysis is a more flexible approach for evaluation of system reliability. A MSS and its components are allowed having more than two levels of working efficiency. These levels are interpreted as states of reliability of the system and its components. However, when only consequences of the failure have to be identified, then BSSs are more suitable for this task. In what follows, we assume that the system is presented as a BSS.

Principal measures of BSSs are reliability function that defines probability that system is functional. There are other measures in reliability analysis. Importance measures are one of groups of these measures [4]. Principal goal of the importance analysis is the investigation of influence of different components states changes on the system

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_49,

reliability. Researchers have developed various methods to calculate importance indices. For example, Markov processes are used to analyze the system state transition process [5] or the structure function approach is used to investigate the system topology [6]. We propose another method that is based on system description by the structure function. This function defines the correlation between system reliability and its components states. The mathematical background of this method is Logical Differential Calculus. This mathematical approach investigates the influence of the variables values on the function value [7].

Two importance measures are considered in this paper. *Component Dynamic Reliability Indices* (CDRIs) allow measuring an influence of each individual component or a fixed group of components on the system reliability. Another importance measure that is very often used in reliability analysis is the *Birnbaum's Importance Measure* (BIM), which defines the probability that given component is critical for system operation [4]. The calculation of these measures for one system component using Logical Differential Calculus has been considered in [8, 9].

In this paper, we develop the investigation of the influence of state changes of more than one component on the system reliability by Direct Partial Logic Derivatives (as the part of Logical Differential Calculus). System failure and its repair caused by changes of some components states are defined in Direct Partial Logic Derivative terminology for two variants of components states changes: for simultaneous and successive changes in the states of components group.

2 Direct Partial Logic Derivatives in Reliability Analysis

2.1 Direct Partial Logic Derivatives

Mathematical approach of Logical Differential Calculus has been introduce for the investigation of the Boolean function value changes depending on the changes of the variables of this function [7]. Therefore, Logical Differential Calculus can be used in applications, where an investigated object is defined and presented by the Boolean function. One of such application problems is reliability analysis. As a rule in reliability analysis, the investigated object is defined as a system with two states that allows analyzing the condition of the system failure and functioning [6, 10]. Such system is named as a *Binary-State System* (BSS).

Consider the system of n components with states x_i ($i = 1,2,..., n$). A system in stationary state will be considered in this paper. At that, the value $x_i = 1$ corresponds to the operable state of the i-th component and $x_i = 0$ to its failure. The correlation of the system state in the fixed time (system availability) and components states is defined by the structure function $\phi(\boldsymbol{x})$:

$$\phi(x_1, x_2, \dots, x_n) = \phi(\boldsymbol{x}): \ \{0,1\}^n \to \{0,1\}, \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ is a state vector.

Every system component is characterized by probabilities of its working and failed state:

$$p_i = \Pr\{x_i = 1\},\ \ q_i = \Pr\{x_i = 0\},\ \ p_i + q_i = 1. \tag{2}$$

The system availability A and unavailability U are defined based on the structure function in the following way:

$$A = \Pr\{\phi(\boldsymbol{x}) = 1\},\ U = \Pr\{\phi(\boldsymbol{x}) = 0\},\ \ A + U = 1. \tag{3}$$

In reliability analysis, the following assumptions are used for the system that is coherent [4]:

(*a*) The structure function can be interpreted as Boolean function,
(*b*) The structure function is monotone,
(*c*) All components are *s*-independent and are relevant to the system.

The assumption (*a*) is very important because it allows using the mathematical approaches of Boolean algebra for the investigation of the system availability. One of such approaches is Logical Differential Calculus [7].

The structure function (1) is a Boolean function; therefore, the mathematical approach of Logical Differential Calculus can be used for the analysis of the influence of component state changes on the system availability. Direct Partial Logic Derivatives are part of Logic Differential Calculus. These derivatives reflect the change in the value of the underlying function when the values of variables change and can be applied for analysis of dynamic behavior of a system that is declared by the structure function (1) according to the assumption (a).

A Direct Partial Logic Derivative with respect to variables vector for the structure function allows estimating the change of system reliability caused by state changes of some system components. These components are interpreted as the vector of components. A Direct Partial Logic Derivative with respect to variables vector for the structure function permits to analyze the system availability change when values of every variable of this vector changes from $\boldsymbol{a}^{(m)}$ to $\overline{\boldsymbol{a}^{(m)}}$. The Direct Partial Logic Derivative of the structure function $\phi(\boldsymbol{x})$ of n variables with respect to vector of components $\boldsymbol{x}^{(m)} = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})$ is defined as follows [10]:

$$\partial\phi(j \to \bar{\jmath})/\partial\boldsymbol{x}^{(m)}\left(\boldsymbol{a}^{(m)} \to \overline{\boldsymbol{a}^{(m)}}\right) = \begin{cases} 1, & \text{if } \phi\left(\boldsymbol{a}^{(m)}, \boldsymbol{x}\right) = j \text{ AND } \left(\overline{\boldsymbol{a}^{(m)}}, \boldsymbol{x}\right) = \bar{\jmath} \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where $\phi\left(\boldsymbol{a}^{(m)}, \boldsymbol{x}\right) = \phi\left(a_{i_1}, a_{i_2}, \ldots, a_{i_m}, \boldsymbol{x}\right)$ is the value of the structure function, when $x_{i_1} = a_{i_1},\ x_{i_2} = a_{i_2}, \ldots,\ x_{i_m} = a_{i_m}$ and $\phi\left(\overline{\boldsymbol{a}^{(m)}}, \boldsymbol{x}\right) = \phi\left(\overline{a_{\imath_1}}, \overline{a_{\imath_2}}, \ldots, \overline{a_{\imath_m}}, \boldsymbol{x}\right)$ is the value of the structure function, when $x_{i_1} = \overline{a_{\imath_1}},\ x_{i_2} = \overline{a_{\imath_2}}, \ldots,\ x_{i_m} = \overline{a_{\imath_m}}$.

Equation (4) for $m = 1$ is Direct Partial Logic Derivative $\partial\phi(j \to \bar{\jmath})/\partial x_i(a \to \bar{a})$ of the structure function $\phi(\boldsymbol{x})$ with respect to variable x_i. This derivative reflects the fact of changing of function from j to $\bar{\jmath}$ when the value of the variable x_i is changing from a to $\bar{a}$:

$$\partial\phi(j \to \bar{\jmath})/\partial x_i(a \to \bar{a}) = \begin{cases} 1, & \text{if } \phi(a_i, \boldsymbol{x}) = j \text{ AND } \phi(\bar{a}_\imath, \boldsymbol{x}) = \bar{\jmath} \\ 0, & \text{in other cases} \end{cases}, \tag{5}$$

where $\phi(a_i, \boldsymbol{x}) = \phi(x_1, \ldots, x_{i\text{-}1}, a, x_{i+1}, \ldots, x_n)$ and $\phi(\bar{a}_\imath, \boldsymbol{x}) = \phi(x_1, \ldots, x_{i\text{-}1}, \bar{a}, x_{i+1}, \ldots, x_n)$.

Direct Partial Logic Derivative (4) is a mathematical description of the system reliability change depending on state changes of the fixed system components. This derivative is used to investigate system reliability change when states of the fixed system components change.

2.2 System Failure

Consider the system failure and repair depending on some system components states changes in terms of Direct Partial Logic Derivatives. There are two types of changes of components states for fixed system components: simultaneous states changes and state changes one by one.

The first variant, the simultaneous state changes, is represented using Direct Partial Logic Derivative terminology as follows [11, 12]:

$$\partial\phi(1 \to 0)/\partial \boldsymbol{x}^{(m)}\left(\mathbf{1}^{(m)} \to \mathbf{0}^{(m)}\right), \tag{6}$$

where $\mathbf{1}^{(m)} = \left(1_{i_1}, 1_{i_2}, \ldots, 1_{i_m}\right)$ and $\mathbf{0}^{(m)} = \left(0_{i_1}, 0_{i_2}, \ldots, 0_{i_m}\right)$.

The assumption (b) that the structure function is monotone is used in (6). Therefore, the system failure is declared by change of function $\phi(\boldsymbol{x})$ from state 1 to state 0 and only decreases of every of m system components from state 1 to state 0 is taken into account. This assumption is also used in the second variant of mathematical description of the system failure.

The second variant of the system failure (system components fail one by one, if they can) is represented as m-times Direct Partial Logic Derivative:

$$\partial^m\phi(1 \to 0)/\partial x_{i_1}(1 \to 0)\partial x_{i_2}(1 \to 0) \ldots \partial x_{i_m}(1 \to 0). \tag{7}$$

The m-times derivative (7) is calculated by successive computation of Direct Partial Logic Derivatives with respect to variable x_{i_s}, for $s = 1,2,\ldots, m$:

$$\partial\phi_{s-1}(1 \to 0)/\partial x_{i_s}(1 \to 0),$$

where the initial function for s-step is defined as $\phi_s(\boldsymbol{x}) = \phi\left(0_{i_1}, 0_{i_2}, \ldots, 0_{i_s}, \boldsymbol{x}\right)$ and $\phi_0(\boldsymbol{x}) = \phi(\boldsymbol{x})$.

For example, consider the failure of system that is presented in Fig. 1. This system consists of three components ($n = 3$) and its structure function is defined as $\phi(\boldsymbol{x}) = \mathrm{OR}(\mathrm{AND}(x_1, x_2), x_3)$. The two aforementioned variants of the failure of this system when component 1 or 2 failed are presented in Table 1, where symbol '-' means that the Direct Partial Logic Derivative does not exist.

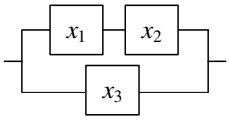


Fig. 1. Series-parallel system

In Table 1, Direct Partial Logic Derivative $\partial\phi(1 \to 0)/\partial \boldsymbol{x}^{(2)}\big((1_1, 1_3) \to (0_1, 0_3)\big)$ models situations, in which the simultaneous breakdown of the 1-st and the 3-rd component causes the system failure. In this case, the system has two boundary states for components $x_1x_2x_3$: {101, 111}. The second Direct Partial Logic Derivative $\partial^2\phi(1\to0)/\partial x_3(1\to0)\partial x_1(1\to0)$ describes system failure when the 3-rd component breaks firstly and then the 1-st component (if can, i.e. if the first component is not failed). The system for this variant of the failure has the following boundary states $x_1x_2x_3$: {001, 011, 101, 111}. The last Direct Partial Logic Derivative $\partial^2\phi(1\to0)/\partial x_1(1\to0)\partial x_3(1\to0)$ describes system failure when component 1 breaks as first and component 3 as the next one. There are three situations $x_1x_2x_3$: {110, 101, 111} in which successive failures of the first and the third components cause the system breakdown (if the third component can fail).

Table 1. The structure function of the system in Fig. 1 and two variants of its failure

x_1	x_2	x_3	$\phi(\boldsymbol{x})$	$\frac{\partial\phi(1 \to 0)}{\partial \boldsymbol{x}^{(2)}\big((1_1, 1_3) \to (0_1, 0_3)\big)}$	$\frac{\partial^2\phi(1 \to 0)}{\partial x_3(1 \to 0)\partial x_1(1 \to 0)}$	$\frac{\partial^2\phi(1 \to 0)}{\partial x_1(1 \to 0)\partial x_3(1 \to 0)}$
0	0	0	0	-	-	-
0	0	1	1	-	1	-
0	1	0	0	-	-	-
0	1	1	1	-	1	-
1	0	0	0	-	-	0
1	0	1	1	1	1	1
1	1	0	1	-	-	1
1	1	1	1	1	1	1

2.3 System Repair

The system repair, using Direct Partial Logic Derivatives, has been defined for state change of one system component in [10] and for state changes of fixed system components in [11]:

$$\partial\phi(0 \to 1)/\partial \boldsymbol{x}^{(m)}\big(\mathbf{0}^{(m)} \to \mathbf{1}^{(m)}\big), \tag{8}$$

where $\mathbf{0}^{(m)} = \big(0_{i_1}, 0_{i_2}, \dots, 0_{i_m}\big)$ and $\mathbf{1}^{(m)} = \big(1_{i_1}, 1_{i_2}, \dots, 1_{i_m}\big)$.

The system repair in Direct Partial Logic Derivative terminology (8) is declared as the structure function change from value 0 into 1, when states of m failed system components change from 0 into 1.

Here, we present state changes of fixed system components one by one for system repair in Direct Partial Logic Derivative terminology as m-times Direct Partial Logic Derivative:

$$\partial^m\phi(0 \to 1)/\partial x_{i_1}(0 \to 1)\partial x_{i_2}(0 \to 1) \dots \partial x_{i_m}(0 \to 1). \tag{9}$$

The computation of derivative (9) is similar to calculation of derivative (7).

3 Importance Measures

Importance measures are very often used in reliability analysis. They estimate coincidence between component failure (repair) and system failure (repair). In this section, we generalize some of them for the simultaneous or successive failure (repair) of components group.

3.1 Component Dynamic Reliability Indices

In this paper, the concept of the CDRIs is generalized for the reliability changes that are caused by state changes (simultaneous or successive) of a group of system components. The basic mathematic model for these indices is based on (6) – (9).

Definition 1. The CDRI of the first type of a group of m components for the system failure is probability of the system failure caused by simultaneous failure of given system components:

$$P_{1f}\left(\boldsymbol{x}^{(m)}\right) = \frac{\rho_{1f}}{\rho_1}\prod_{j=1}^{m} q_{i_j}, \tag{10}$$

where ρ_{1f} is a number of situations when the breakdown of m system components results the failure of the system; ρ_1 is a number of operational system states when $\phi\left(1_{i_1}, 1_{i_2}, \ldots, 1_{i_m}, \boldsymbol{x}\right) = 1$ (it is computed by the structure function); and q_{i_j} is the failed state of component i_j (2).

Number ρ_{1f} is calculated as a number of nonzero elements of Direct Partial Logic Derivative (6):

$$\rho_{1f} \equiv \partial\phi(1 \to 0)/\partial \boldsymbol{x}^{(m)}\left(\mathbf{1}^{(m)} \to \mathbf{0}^{(m)}\right) \neq 0. \tag{11}$$

Definition 2. The CDRI of the second type of a group of m components for the system failure is probability of the system failure caused by successive failure of given system components:

$$P_{2f}\left(\boldsymbol{x}^{(m)}\right) = \sum_{s=1}^{m} \frac{\rho_{f,i_s}}{\rho_{1,i_s}} p_{i_s} \prod_{j=1}^{s-1} q_{i_j}, \tag{12}$$

where ρ_{1,i_s} is a number of system states when $x_{i_1} = x_{i_2} = \cdots = x_{i_s} = 1$ and $\phi(\boldsymbol{x}) = 1$; q_{i_j} is the i_j-th component unreliability and p_{i_s} is the i_s-th component reliability according to (2); ρ_{f,i_s} is a number of system states for which breakdown of s components, from the i_1-th to the i_s-th, cause system failure and it is calculated by the structure function as a number of nonzero elements of Direct Partial Logic Derivative (7).

Definition 3. The CDRI of the first type of a group of m components for the system repair is probability of the system repair caused by simultaneous replacements of m failed system components:

$$P_{1r}\left(\boldsymbol{x}^{(m)}\right) = \frac{\rho_{1r}}{\rho_0}\prod_{j=1}^{m} p_{i_j}, \tag{13}$$

where ρ_{1r} is a number of system states when the simultaneous replacements of m failed system components causes the system repair and it is calculated by Direct Partial Logic Derivative (8):

$$\rho_{1r} \equiv \partial\phi(0 \to 1)/\partial \boldsymbol{x}^{(m)}\left(\mathbf{0}^{(m)} \to \mathbf{1}^{(m)}\right) \neq 0, \tag{14}$$

and ρ_0 is a number of zero system states for which $\phi\left(0_{i_1}, 0_{i_2}, \ldots, 0_{i_m}, \boldsymbol{x}\right) = 0$ and q_{ij} is the probability that component i_j is failed (2).

Definition 4. The CDRI of the second type of a group of m components for the system repair is probability of system repair caused by successive replacements of m failed system components:

$$P_{2r}\left(\boldsymbol{x}^{(m)}\right) = \sum_{s=1}^{m} \frac{\rho_{r,i_s}}{\rho_{0,i_s}} q_{i_s} \prod_{j=1}^{s-1} p_{i_j}, \tag{15}$$

where $\rho_{0,is}$ is a number of situations when $x_{i_1} = x_{i_2} = \cdots = x_{i_s} = 0$ and $\phi(\boldsymbol{x}) = 0$; p_{i_j} is the i_j-th component reliability and q_{i_s} is the i_s-th component unreliability according to (2); ρ_{r,i_s} is a number of system states for which replacements of system components, from the i_1-th to the i_s-th, cause the system repair and ρ_{r,i_s} is calculated by the system structure function and Direct Partial Logic Derivative (9) as a number of its nonzero elements.

3.2 The Birnbaum's Importance Measure

The *Birnbaum's Importance Measure* (BIM) defines probability that given component is critical for system operation, i.e. its failure causes the system failure [4]. In papers [8, 11], there has been proposed the definition of the BIM for component i as the probability that Direct Partial Logic Derivative $\partial\phi(1 \to 0)/\partial x_i(1 \to 0)$ contains nonzero values:

$$BIM(x_i) = \Pr\{\partial\phi(1 \to 0)/\partial x_i(1 \to 0) = 1\}. \tag{16}$$

Generalizations of the BIM for two system components have been considered in paper [13]. Those generalizations are known as Joint Reliability (Failure) Importance Measures and they represent the degree of interactions between two system components. However, they do not estimate the probability that simultaneous or successive failures of some components cause the system failure. Therefore, in this paper, the definition of the BIM is generalized for these cases. We denote these generalizations as the *Modified BIM* (MBIM).

Definition 5. The MBIM of the first type of a group of m components for the system failure is probability that the simultaneous failure of the fixed group of system components results the failure of the system:

$$MBIM_1\left(\boldsymbol{x}^{(m)}\right) = \Pr\left\{\partial\phi(1 \to 0)/\partial\boldsymbol{x}^{(m)}\left(\mathbf{1}^{(m)} \to \mathbf{0}^{(m)}\right) = 1\right\}. \quad (17)$$

Definition 6. The MBIM of the second type of a group of m components for the system failure is probability that the successive failures of the fixed system components cause the system failure:

$$MBIM_2\left(\boldsymbol{x}^{(m)}\right) = \sum_{s=1}^{m} \Pr\left\{\frac{\partial\phi_{s-1}(1\to 0)}{\partial x_{is}(1\to 0)} = 1\right\} \prod_{j=1}^{s-1} \Pr\left\{\frac{\partial\phi_{j-1}(1\to 0)}{\partial x_{ij}(1\to 0)} = 0\right\}, \quad (18)$$

where $\phi_s(\boldsymbol{x}) = \phi\left(0_{i_1}, 0_{i_2}, \ldots, 0_{i_s}, \boldsymbol{x}\right)$ and $\phi_0(\boldsymbol{x}) = \phi(\boldsymbol{x})$. The assumption (*c*) for the structure function, i.e. all components are independent and relevant to the system, must hold for the definition (18) of the MBIM of the second type.

3.3 Examples for Calculation of the CDRIs and the BIM

Consider the example in Fig. 1 and calculation of the CDRIs and the MBIM for it. Let the probabilities of the states of components be the ones in Table 2. The structure function of this system is:

$$\phi(\boldsymbol{x}) = \mathrm{OR}(\mathrm{AND}(x_1, x_2), x_3). \quad (19)$$

Let us calculate the probabilities of this system failure if two components of it fail. The CDRIs for this system failure are determined according to (10) and (12).

Table 2. State probabilities of components of the system in Fig. 1

	Components		
	x_1	x_2	x_3
q_i	0.03	0.12	0.03
p_i	0.97	0.88	0.97

The CDRIs of the first type $P_{1f}(\boldsymbol{x}^{(2)})$ (for two simultaneous components breakdown) for this system are: $P_{1f}(x_1, x_2) = 0.4268$, $P_{1f}(x_1, x_3) = 0.9409$ and $P_{1f}(x_2, x_3) = 0.8536$.

Numbers ρ_{1f} for different changes of system components (11) are determined by derivatives $\partial\phi(1 \to 0)/\partial\boldsymbol{x}^{(2)}\left(\left(1_{i_1}, 1_{i_2}\right) \to \left(0_{i_1}, 0_{i_2}\right)\right)$ and number ρ_1 is computed by the structure function of this system.

The CDRIs of the second type for the system in Fig. 1 are calculated for successive components breakdowns by (12) and this equation in this case is:

$$P_{2f}\left(\boldsymbol{x}^{(2)}\right) = \frac{\rho_{f,i_1}}{\rho_{1,i_1}} p_{i_1} + \frac{\rho_{f,i_2}}{\rho_{1,i_2}} q_{i_1} p_{i_2}, \quad (20)$$

where ρ_{1,i_1} is a number of system states where $x_{i_1} = 1$ and $\phi(\boldsymbol{x}) = 1$; ρ_{1,i_2} is a number of system states where $x_{i_1} = x_{i_2} = 1$ and $\phi(\boldsymbol{x}) = 1$; ρ_{f,i_1} is a number of state vectors for which the i_1-th component breakdown causes system failure and ρ_{f,i_2} is a number of

state vectors for which the i_1-th and the i_2-th components breakdowns cause system failure; p_{i_1} (q_{i_1}) and p_{i_2} are the i_1-th and the i_2-th components reliabilities (unreliability) according to (2).

Numbers ρ_{1,i_1} and ρ_{1,i_2} in (20) are computed by structure function (19) of the system. Numbers ρ_{1,i_1} and ρ_{1,i_2} are determined by Direct Partial Logic Derivative (7) as a number of its nonzero elements.

The CDRIs of the second type $P_{2f}(\boldsymbol{x}^{(2)})$ for this system are calculated for successive components breakdowns by (20) as follows: $P_{2f}(x_1, x_2) = 0.3233$, $P_{2f}(x_2, x_1) = 0.2933$, $P_{2f}(x_1, x_3) = 0.3524$, $P_{2f}(x_3, x_1) = 0.7421$, $P_{2f}(x_2, x_3) = 0.4097$ and $P_{2f}(x_3, x_2) = 0.7407$. These numbers show that the most probable scenarios which lead into the failure of the system are successive failures of the 3-rd and the 1-st component or successive failures of components 3 and 2.

The MBIM of the first type (for two simultaneous components breakdown) for this system are: $MBIM_1(x_1, x_2) = 0.03$, $MBIM_1(x_1, x_3) = 1$ and $MBIM_1(x_2, x_3) = 1$. These numbers mean that the simultaneous failure of component 1 and 3 or 2 and 3 causes the failure of the system regardless to state of another component.

The MBIM of the second type are computed for this system according to (18) as follows:

$$MBIM_2\left(\boldsymbol{x}^{(2)}\right) = \Pr\left\{\frac{\partial\phi(1\to 0)}{\partial x_{i1}(1\to 0)} = 1\right\} + \Pr\left\{\frac{\partial\phi(1\to 0)}{\partial x_{i1}(1\to 0)} = 0\right\}\Pr\left\{\frac{\partial\phi_1(1\to 0)}{\partial x_{i2}(1\to 0)} = 1\right\}, \quad (21)$$

where $\phi_1(\boldsymbol{x}) = \phi\left(0_{i_1}, \boldsymbol{x}\right)$.

Using (21), we get the next values of the MBIM of the second type for the system in Fig. 1: $MBIM_2(x_1, x_2) = 0.0264$, $MBIM_2(x_1, x_3) = 1$ and $MBIM_2(x_2, x_3) = 1$. These numbers mean that the successive failure of component 1 and 3 or 2 and 3 causes the failure of the system every time, i.e. regardless to state of another component.

4 Conclusion

Two new importance measures are proposed in this paper: extension of the CDRI and the BIM. The CDRI is used to investigate the system availability depending on the components states changes. New type of the BIM has been denoted as the Modified BIM, because it does not estimate the overall criticality of given group of components for the system activity but only the influence of their simultaneous or successive failures.

The novelty of the proposed measures consists of two aspects. One of them is investigation of influence of some fixed components states changes on the system availability. As a rule, importance measures are defined for only one system component. In this work, the simultaneous or successive components states changes are considered.

Another aspect of the novelty is using of the Direct Partial Logic Derivatives with respect to the vector of variables for calculation of these measures. The advantages of this mathematical approach are: (a) the application for analysis of system with any complexity and (b) the simplicity of the calculation. The computational complexity of

the Direct Partial Logic Derivatives calculation depends on the number of system components only and is not influenced by other topological complexity of the system. The Direct Partial Logic Derivatives are calculated based on the comparison of the values of the structure function only, and therefore, their calculation is very simple.

Logical Differential Calculus is very perspective for analysis of dynamic properties of the investigated system. Its potential in reliability analysis can increase in combination with Logical Integral Calculus [14] that allows revealing the structure function when it is not completely defined. However, this idea needs another research.

Acknowledgement. This work was partially supported by the grant of Slovak Research and Development Agency SK-PL-0023-12 and the grant of 7th RTD Framework Program No 610425 (RASimAs).

References

1. Lisnianski, A., Levitin, G.: Multi-State System Reliability. Assessment, Optimization and Applications. World Scientific (2003)
2. Xie, M., Dai, Y.-S., Poh, K.-L.: Multi-State System Reliability. In: Xie, M., Dai, Y.-S., Poh, K.-L. (eds.) Computing System Reliability. Models and Analysis, pp. 207–237. Kluwer Academic Publishers (2004)
3. Shooman, M.L.: Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design. John Wiley & Sons, Inc. (2002)
4. Kuo, W., Zhu, X.: Importance Measures in Reliability, Risk and Optimization. John Wiley & Sons, Ltd. (2012)
5. Fricks, R.M., Trivedi, K.S.: Importance analysis with Markov chains. In: Proc. of Reliability and Maintainability Annual Symposium, pp. 89–95 (2003)
6. Armstrong, M.J.: Reliability-importance and dual failure-mode components. IEEE Trans. Reliability 46, 212–221 (1997)
7. Akers, S.B.: On a Theory of Boolean Functions. J. Soc. Ind. Appl. Math. 7, 487–498 (1959)
8. Zaitseva, E., Levashenko, V.: Importance Analysis by Logical Differential Calculus. Automation and Remote Control 74(2), 171–182 (2013)
9. Moret, B.M.E., Thomason, M.G.: Boolean Difference Techniques for Time-Sequence and Common-Cause Analysis of Fault-Trees. IEEE Trans Reliability R-33, 399–405 (1984)
10. Zio, E.: An Intriduction to the Basics of reliability and Risk Analysis. World Scientific (2007)
11. Zaitseva, E., Puuronen, S.: Estimation of Multi-State system reliability depending on changes of some system component efficiencies. In: Proc. of European Safety and Reliability Conference (ESREL 2007), pp. 253–261. Taylor & Francis Group (2007)
12. Kvassay, M., Levashenko, V.: Birnbaum Importance for Estimation of Multi-state and Binary-state Systems. Radioelectronic and Computer Systems 64(5), 261–266 (2013)
13. Hong, J.S., Lie, C.H.: Joint Reliability-importance of Two Edges in an Undirected Network. IEEE Trans. on Reliability 42(1), 17–23, 33 (1993)
14. Tucker, J.H., Tapia, M.A., Bennett, A.W.: Boolean Integral Calculus for Digital Systems. IEEE Trans. on Computers C-34(1), 78–81 (1985)

Model Fusion for the Compatibility Verification of Software Components

W.M. Zuberek

Department of Computer Science, Memorial University,
St.John's, NL, Canada A1B 3X5
wlodek@mun.ca

Abstract. Similarly as in earlier work on component compatibility, the behavior of components is specified by component interface languages, defined by labeled Petri nets. In the case of composition of concurrent components, the requests from different components can be interleaved, and - as shown earlier - such interleaving can result in deadlocks in the composed system even if each pair of interacting components is deadlock–free. Therefore the elements of a component–based system are considered compatible only if the composition is deadlock–free. This paper formally defines model fusion, which is a composition of net models of individual components that represents the interleaving of interface languages of interacting components. It also shows that the verification of component compatibility can avoid the exhaustive analysis of the composed state space.

Keywords: software components, component-based systems, component composition, component compatibility, compatibility verification, model fusion, labelled Petri nets.

1 Introduction

In component-based software development the functionality of a software ssystem is decomposed among loosely coupled independent software components. Such a reuse-oriented approach to defining and implementing software systems has recently been extensively studied as it is believed to be a starting platform for service orientation [4], [10].

In component-based systems [10], two interacting components, one requesting services and the other providing them, are considered compatible if all possible sequences of services needed by the requesting component can be provided by the other one. This concept of component compatibility can be extended to sets of interacting components, however, in the case of several concurrent requester components, as is typically the case for client–server applications, the requests from different components can be interleaved and then verifying component compatibility must take into account all possible interleavings of requests. Such interleaving of requests can lead to unexpected behavior of the composed system, i.e. a deadlock can occur indicating component incompatibility [21], [22].

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX,* Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_50,

The behavior of components is usually described at component interfaces [17] and the components are characterized as requester (active) and provider (reactive) components. Although several approaches to checking component composability have been proposed [1], [2], [4], [11], [13], [18], further research is needed to make these ideas practical [9].

The paper is an extension of previous work on component compatibility and substitutability [8], [20], [21], [22]. Using the same formal specification of component behavior in the form of interface languages, the paper addresses the verification of component compatibility. Since interface languages are usually infinite, their compact finite specification is needed for effective processing. Labeled Petri nets [20], [21] are used as such specification.

Petri nets [14], [15] are formal models of systems which exhibit concurrent activities with constraints on frequency or orderings of these activities. In labeled Petri nets, labels, which represent services, are associated with elements of nets in order to control component interactions. Well-developed mathematical theory of Petri nets provides a convenient formal foundation for analysis of systems modeled by Petri nets.

Model fusion is proposed to represent the interactios of components. The fusion operation is performed by merging Petri net transitions that request and provide the same service. Interleavings of requests from concurrent components result in different execution paths in the combined model. The components are compatible, if the combined model is deadlock–free. Deadlock–freeness can be verified by structural methods, avoiding the exhaustive state space analysis because model fusion preserves structural (some) properties of the fused elements. If the interacting components are not compatible, some correcting steps are required (in the form of redesign of some components or additional constraints which prevent some interleavings to occur).

Section 2 recalls the concept of interface languages as the description of component's behavior and Section 3 presents a linguistic version of component compatibility. Section 4 illustrates the proposed concepts by a simple example and shows that compatibility verification for some classes of systems can be performed using structural analysis of the fused model. Section 5 concludes the paper.

2 Component Behavior

The behavior of a component, at its interface, can be represented by a cyclic labeled Petri net [7], [8], [21]:

$$\mathcal{M}_i \;=\; (P_i, T_i, A_i, S_i, m_i, \ell_i, F_i),$$

where P_i and T_i are disjoint sets of places and transitions, respectively, A_i is the set of directed arcs, $A_i \subseteq P_i \times T_i \cup T_i \times P_i$, S_i is an alphabet representing the set of services that are associated with transitions by the labeling function $\ell_i : T_i \to S_i \cup \{\varepsilon\}$ (ε is the "empty" service; it labels transitions which do not

represent services), m_i is the initial marking function $m_i : P_i \to \{0, 1, ...\}$, and F_i is the set of final markings (which are used to capture the cyclic nature of sequences of firings).

Sometimes it is convenient to separate net structure $\mathcal{N} = (P, T, A)$ from the initial marking function m.

In order to represent component interactions, the interfaces are divided into *provider* interfaces (or p-interfaces) and *requester* interfaces (or r-interfaces). In the context of a provider interface, a labeled transition can be thought of as a service provided by that component; in the context of a requester interface, a labeled transition is a request for a corresponding service. For example, the label can represent a conventional procedure or method invocation. It is assumed that if the p-interface requires parameters from the r-interface, then the appropriate number and types of parameters are delivered by the r-interface. Similarly, it is assumed that the p-interface provides an appropriate return value, if such a value is required. The equality of symbols representing component services (provided and requested) implies that all such requirements are satisfied.

For unambiguous interactions of requester and provider interfaces, it is required that in each p-interface there is exactly one labeled transition for each provided service:

$$\forall t_i, t_j \in T : \ell(t_i) = \ell(t_j) \neq \varepsilon \Rightarrow t_i = t_j.$$

Moreover, to express the reactive nature of provider components, all provider models are required to be ε–conflict–free, *i.e.*:

$$\forall t \in T \; \forall p \in Inp(t) : Out(p) \neq \{t\} \Rightarrow \ell(t) \neq \varepsilon$$

where $Out(p) = \{t \in T \mid (p, t) \in A\}$; the condition for ε–conflict–freeness could be used in a more relaxed form but this is not discussed here for simplicity of presentation.

Component behavior is determined by the set of all possible sequences of services (required or provided by a component) at a particular interface. Such a set of sequences is called the *interface language*.

Let $\mathcal{F}(\mathcal{M})$ denote the set of firing sequences in $\mathcal{M}$ such that the marking created by each firing sequence belongs to the set of final markings F of $\mathcal{M}$. The interface language $\mathcal{L}(\mathcal{M})$, of a component represented by a labeled Petri net $\mathcal{M}$, is the set of all labeled firing sequences of $\mathcal{M}$:

$$\mathcal{L}(\mathcal{M}) \; = \; \{\ell(\sigma) \mid \sigma \in \mathcal{F}(\mathcal{M})\},$$

where $\ell(t_{i_1} t_{i_2} ... t_{i_k}) = \ell(t_{i_1})\ell(t_{i_2})...\ell(t_{i_k})$.

By using the concept of final markings, interface languages can easily capture the cyclic behavior of (requester as well as provider) components.

Interface languages defined by Petri nets include regular languages, some context–free and even context–sensitive languages [12]. Therefore, they are significantly more general than languages defined by finite automata [5], but their compatibility verification is also more difficult than in the case of regular languages.

3 Component Compatibility

Interface languages of interacting components are used to define the compatibility of components. A pair of interacting components, a requester component "r" and a provider component "p", are compatible if and only if all sequences of services requested by "r" can be provided by "p", i.e., if and only if:

$$\mathcal{L}_r \subseteq \mathcal{L}_p.$$

In the case of several requester components, "r_i", $i \in I$, interacting with a single provider component "p", the component compatibility requires that all sequences of (interleaved) requests be satisfied by the provider, so:

$$\mathcal{L}_I \subseteq \mathcal{L}_p$$

where $\mathcal{L}_I$ is the language of interleavings of requester languages $\mathcal{L}_i, i \in I$. It should be observed that $\mathcal{L}_I$ does not necessarily contain all possible interleavings of requests because some requests cannot be satisfied immediatedy upon request and are delayed until some other operations and/or their sequences are performed by the provider component. All such restrictions are represented by the fused component models.

4 Model Fusion

Model fusion is used to combine separate models of interacting components into one model which represents the possible behaviors of the interacting components. The general idea is sketched in Fig.1 where a single transition representing service "a" in the provider component is combined with requests of this service in two requester components - the provider transition is conceptually "fused" (or merged) with corresponding transitions of the requester components.

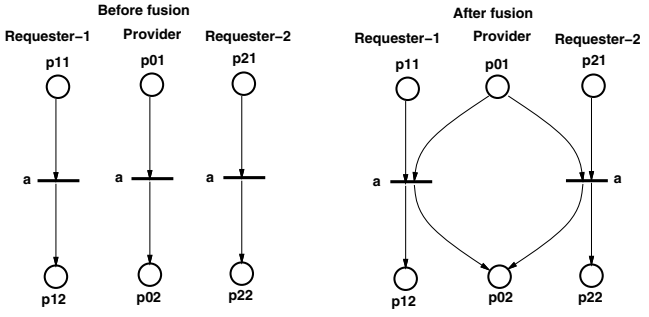


Fig. 1. Model fusion for service "a"

More formally, if the provider component is modeled by

$$\mathcal{M}_p \;=\; (P_p, T_p, A_p, S, m_p, \ell_p, F_p),$$

and the requester components are modeled by

$$\mathcal{M}_i \;=\; (P_i, T_i, A_i, S, m_i, \ell_i, F_i), \quad i \in I,$$

where S is a common set of services and all other sets (of places, transitions, etc.) are disjoint, then the combined (or "fused") model is

$$\mathcal{M}_c \;=\; (P_c, T_c, A_c, S, m_c, \ell_c, F_c),$$

where:

$$P_c = P_p \cup P_I, \quad P_I = \bigcup_{i\in I} P_i;$$

$$T_c = T_p - T_0 \cup T_I. \;\; T_0 = \{t \in T_p \mid \ell_p(t) \in S\}, \quad T_I = \bigcup_{i\in I} T_i;$$

$$A_c = A_p - A_0 \cup A_I \cup A_{pr}, \quad A_0 = P_c \times T_0 \cup T_0 \times P_c, \quad A_I = \bigcup_{i\in I} A_i,$$

$$A_{pr} = \{(p_x, t_{ik}) \mid p_x \in P_p \wedge (p_x, t) \in A_0 \wedge \ell_i(t_{ik}) = \ell_p(t)\} \cup$$

$$\{(t_{ik}, p_y) \mid p_y \in P_p \wedge (t, p_y) \in A_0 \wedge \ell_i(t_{ik}) = \ell_p(t)\};$$

$$\forall p \in P_c : m(p) = \begin{cases} m_p(p), & \text{if } p \in P_p, \\ m_i(p), & \text{if } p \in P_i, i \in I; \end{cases}$$

$$\forall t \in T_c : \ell(t) = \begin{cases} \ell_p(t), & \text{if } t \in T_p. \\ \ell_i(t), & \text{if } t \in T_i, i \in I; \end{cases}$$

$$F_c = F_p \cup F_I, \quad F_I = \bigcup_{i\in I} F_i;$$

and $Inp(t)$ and $Out(t)$ are, respectively, the input and the output sets places of transition t. The composed model is obtained by deleting all labeled transitions in the provider model (the set T_0) with all arcs connected to these transitions (the set A_0), and adding arcs similar to the deleted ones but connecting places of the provider model with labeled transitions of all requester models (the set A_{pr}).

The composed net can be analyzed by typical methods used for analysis of Petri net models. In particular, for some classes of models, structural analysis can be used for verification of deadlock freeness [19], [16] avoding the exhaustive state–based model analysis.

It can be observed that the fusion operation preserves the place invariants [15] of the model, as well as their siphons. More specifically, if a collection of component models is covered by a set of place invariants, than the same set of invariants covers the fused model. Moreover, if all place invariants are marked, than no deadlock can occur in the fused model, so the components are compatible. The following example illustrates this property in greater detail.

5 Example

A simple system of two requesters and a single provider is shown in Fig.2. The interface language of Requester-1 is simply `(ab)*`, the language of Provider describes the behavior of (unbounded) stack with services "`a`" and "`b`" corresponding to operations "push" and "pull', and the languge of Requester-2 is that of boumded stack of capacity 2. Both stacks (i.e., Provider and Requester-2) in Fig.2 are empty. Obviously, the languages of Requester-1 and Requester-2 are (proper) subsets of the language of Provider.

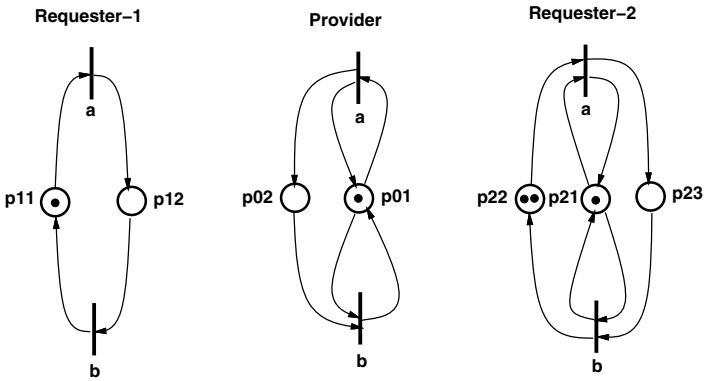


Fig. 2. Net models of two requesters and a single provider

Fig.3 shows the combined (or fused) model of the system from Fig.2. Structural analysis can be used to check its deadlock freeness (i.e., compatibility of components).

It is known that a deadlock in a Petri net corresponds to an unmarked siphon [6] (a siphon is a subset of places for which the set of output transitions is a superset of the set of input transitions). Consequently, a deadlock freeness can be verified by checking that the siphons cannot become unmarked (linear programming can be used to for such checking [19]). Moreover, since all siphons are composed of a rather small number of basis siphons [3], verification can be restricted to basis siphons only.

The net shown in Fig.3 has three such siphons with the following subsets of places:

siphon	*places*
1	p_{11}, p_{12}
2	p_{11}, p_{01}, p_{21}
3	$p_{22}.p_{23}$

Since all these siphons are actually marked place invariants, they cannot become unmarked (place invariants preserve the markings). Consequently, the

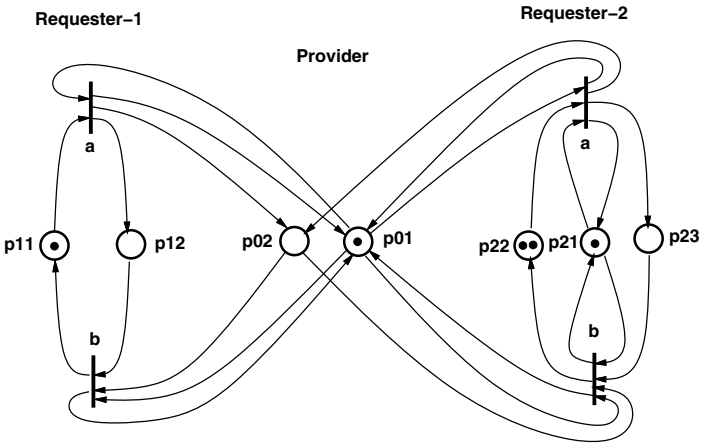


Fig. 3. A combined model of two requesters and a single provider

deadlock cannot occur in the net shown in Fig.3, so the interacting components are compatible.

It can be observed that the original models of components are covered by place invariants (in the example shown in Fig.2, the model of Requester-1 is a single place invariant, the model of Provider is covered by two place invariants, and the model of Requester-2 is also covered by two place invariants). The combined model is covered by the same place invariants because the fusion process preserves the place invariants. Efficient methods of compatibility verification can use such properties for simplifying the verification process.

Morel general structural approach to deadlock analysis is discussed in [19].

6 Concluding Remarks

The paper shows that the verification of component compatibility based on the exhaustive analysis of the "state space", as discussed in [20] and [21], can be replaced by structural analysis of the model obtained by "fusion" of models of interacting components.

It is expected that the proposed verification of component compatibility can be quite efficient although this aspect needs to be studied in greater detail.

It should be noticed that the discussion of component compatibility was restricted to a single provider component. In the case of several providers, each provider can be considered independently of other, so a single provider case is not really a restriction.

Similarly, the requester and provider components can use other components in a sort of hierarchical structure. Since model fusion combines all component models into a single net model, there are no restrictions on such structures.

Also, an important aspect of component compatibility is its incremental verification. The approach described in this paper is not incremental (with the

exception of some cases, for example, when all models of all components are covered by place invariants), but may provide a foundation for an incremental approach.

The paper did not address the question of deriving behavioral models of components (which is common to all component-based studies). Such models, at least theoretically, could be derived from formal component specifications, or perhaps could be obtained through analyzing component implementations. Since the component compatibility verification proposed in this paper does not require the use of the underlying component models (they are used only to define the interface languages), these interface languages could also be determined experimentally, by executing the components and collecting the information about the sequences of service requests.

Acknowledgement. The Natural Sciences and Engineering Research Council of Canada partially supported this research through grant RGPIN-8222.

References

1. Attiogbé, J.C., André, P., Ardourel, G.: Checking component composability. In: Löwe, W., Südholt, M. (eds.) SC 2006. LNCS, vol. 4089, pp. 18–33. Springer, Heidelberg (2006)
2. Baier, C., Klein, J., Klüppelholz, S.: Modeling and verification of components and connectors. In: Bernardo, M., Issarny, V. (eds.) SFM 2011. LNCS, vol. 6659, pp. 114–147. Springer, Heidelberg (2011)
3. Boer, E.T., Murata, T.: Generating basis siphons and traps of Petri nets using the sign incidence matrix. IEEE Trans. on Circuits and Systems, I – Fundamental Theory and Applications 41(4), 266–271 (1994)
4. Broy, M.: A theory of system interaction: components, interfaces, and services. In: Interactive Computations: The New Paradigm, pp. 41–96. Springer (2006)
5. Chaki, S., Clarke, S.M., Groce, A., Jha, S., Veith, H.: Modular verification of software components in C. IEEE Trans. on Software Engineering 30(6), 388–402 (2004)
6. Chu, F., Xie, X.: Deadlock analysis of Petri nets using siphons and mathematical programming. IEEE Trans. on Robotics and Automation 13(6), 793–804 (1997)
7. Craig, D.C., Zuberek, W.M.: Compatibility of software components – modeling and verification. In: Proc. Int. Conf. on Dependability of Computer Systems, Szklarska Poreba, Poland, pp. 11–18 (2006)
8. Craig, D.C., Zuberek, W.M.: Petri nets in modeling component behavior and verifying component compatibility. In: Proc. Int. Workshop on Petri Nets and Software Engineering, Siedlce, Poland, pp. 160–174 (2007)
9. Crnkovic, I., Schmidt, H.W., Stafford, J., Wallnau, K.: Automated component-based software engineering. The Journal of Systems and Software 74(1), 1–3 (2005)
10. Garlan, D.: Formal modeling and analysis of software architecture: Components, connectors, and events. In: Bernardo, M., Inverardi, P. (eds.) SFM 2003. LNCS, vol. 2804, pp. 1–24. Springer, Heidelberg (2003)
11. Henrio, L., Kammüller, F., Khan, M.U.: A framework for reasoning on component composition. In: de Boer, F.S., Bonsangue, M.M., Hallerstede, S., Leuschel, M. (eds.) FMCO 2009. LNCS, vol. 6286, pp. 1–20. Springer, Heidelberg (2010)

12. Hopcroft, J., Motwani, E., Ullman, R., Introduction, J.D.: Introduction to automata theory, languages, and computations, 2nd edn. Addison-Wesley (2001)
13. Leicher, A., Busse, S., Süß, J.G.: Analysis of compositional conflicts in component-based systems. In: Gschwind, T., Aßmann, U., Wang, J. (eds.) SC 2005. LNCS, vol. 3628, pp. 67–82. Springer, Heidelberg (2005)
14. Murata, T.: Petri nets: properties, analysis, and applications. Proceedings of the IEEE 77(4), 541–580 (1989)
15. Reisig, W.: Petri nets – an introduction. EATCS Monographs on Theoretical Computer Science, vol. 4. Springer (1995)
16. Reisig, W.: Understanding Petri nets – modeling techniques, analysis methods, case studies. Springer (2013)
17. Szyperski, C.: Component software: beyond object-oriented programming, 2nd edn. Addison-Wesley Professional (2002)
18. Zaremski, A.M., Wang, J.M.: Specification matching of software components. ACM Trans. on Software Engineering and Methodology 6(4), 333–369 (1997)
19. Zuberek, W.M.: Siphon-based verification of component compatibility. In: Proc. 4th Int. Conference on Dependability of Computer Systems (DepCoS 2009), Brunow Palace, Poland, pp. 123–132 (2009)
20. Zuberek, W., Checking, M.: compatibility and substitutability of software components. In: Models and Methodology of System Dependability. Oficyna Wydawnicza Politechniki Wroclawskiej, ch. 14, pp. 175–186 (2010)
21. Zuberek, W.M.: Incremental composition of software components. In: Zamojski, W., Kacprzyk, J., Mazurkiewicz, J., Sugier, J., Walkowiak, T. (eds.) Dependable Computer Systems. AISC, vol. 97, pp. 301–311. Springer, Heidelberg (2011)
22. Zuberek, W.M.: Service renaming in component composition. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Complex Systems and Dependability. AISC, vol. 170, pp. 319–330. Springer, Heidelberg (2012)

Erratum: CDM: A Prototype Implementation of the Data Mining JDM Standard

Piotr Lasek

Chair of Computer Science, University of Rzeszów
ul. Prof. St. Pigonia 1, 35-310 Rzeszów, Poland
lasek@ur.edu.pl

W. Zamojski et al. (eds.), *Proceedings of the Ninth International Conference DepCoS-RELCOMEX*, Advances in Intelligent Systems and Computing 286,
DOI: 10.1007/978-3-319-07013-1_29, © Springer International Publishing Switzerland 2014

DOI 10.1007/978-3-319-07013-1_51

We would like to kindly inform all readers of the paper that it was inspired by Professor Jan Bazan, who had initially drawn our attention to the JDM standard. He, his team, and cooperating researchers are currently working on a data mining library employing JDM interfaces called CommoDM, that is a continuation of previous systems RSES [1] and RoughICE [2, 5], created by the group of Professor Andrzej Skowron from the Warsaw University. Results of experiments performed using a preliminary version of CommoDM library were presented in [3] and [4]. The implementation presented in the paper confirms possible usefulness of JDM as a supporting technology for CommoDM, in the field of data clustering methods.

References

1. Bazan, J., G., Szczuka, M.: The Rough Set Exploration System, *Transactions on Rough Sets*, III, LNCS 3400, 2005, 37–56.
2. Bazan, J. G.: Hierarchical classifiers for complex spatio-temporal concepts, *Transactions on Rough Sets*, IX, LNCS 5390, 2008, 474–750.
3. Bazan, J., G., Bazan-Socha, S., Buregwa-Czuma, S., Pardel, P., W., Sokolowska, B.: Prediction of coronary arteriosclerosis in stable coronary heart disease, *Proceedings of the Fourteen Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2012, Springer-Verlag, Communica- tions in Computer and Information Science, vol. 298, part 9, Springer, 2012, pp. 550-559.
4. Bazan, J., G., Buregwa-Czuma, S., Jankowski A., W.: A Domain Knowledge as A Tool For Improving Classifiers, Fundamenta Informaticae, Volume 127, Number 1-4, p. 495-511 (2013).
5. Bazan, J. G., Bazan-Socha, S., Buregwa-Czuma, S., Pardel, P. W., Skowron, A., Sokolowska, B.: Classifiers Based on Data Sets and Domain Knowledge: A Rough Set Approach, in: *Rough Sets and Intelligent Systems - Professor Zdzislaw Pawlak in Memoriam. Intelligent Systems Reference Library* (A. Skowron, Z. Suraj, Eds.), vol. 43, Springer-Verlag, Berlin Heidelberg, 2013, 93–136.

The original online version for this chapter can be found at
http://dx.doi.org/10.1007/978-3-319-07013-1_29

Author Index