An approach to Service Net modelling

Wojciech Zamojski, Tomasz Walkowiak
Wroclaw University of Technology, Poland
{wojciech.zamojski, tomasz.walkowiak}@pwr.wroc.pl

ABSTRACT

The paper presents the approach to modelling technical systems. The systems are being analysed from the point of services realised by them. Therefore, the system is modelled as a network of interacting services and technical infrastructure required for the service realisation. Examples of systems that could be seen as a service net are the information systems, distribution of electricity or transport systems. Paper, presents the formal model of a service net, accompanied with the Petri Net model of realising tasks in the service net.

Key Words: service nets, modelling, technical systems, dependability, Petri Net

1. Introduction

Network technologies are being developed for many years. Most of large technical systems could be seen as a kind of network, for example: information, transport or electricity distribution systems. Networks are modelled as directed graphs with nodes, in which commodities and information media are being processed, and arcs as communication links (telecommunication channels, roads, pipelines, conveyors, etc.) for media transportation. Resources of networks could be divided into two classes: services (functionality resources) and technical infrastructures (hardware and software resources).

We propose to analyse the network system from the functional and user point of view, focusing on business service realized by a network system [1]. Users of the network system realise some tasks in the system (for example: send a parcel in the transport system or buy a ticket in the internet ticket office). We assume that the main goal, taken into consideration during design and operation, of the network system is to fulfil the user requirements. Which could be seen as some quantitative and qualitative parameters of user tasks.

Network services and technical resources are engaged for task realization and each task needs a fixed list of services which are processed on the base of whole network technical infrastructure or on its part. Different services may be realized on the same technical resources and the same services may be realized on different sets of technical resources. Of course with different values of performance and reliability parameters. The last statement is essential when tasks are realized in the real network system surrounded by unfriendly environment that may be a source of threads and even intentional attacks. Moreover, the real networks are build of unreliable software and hardware components as well.

In [2] authors described basic set of dependability attributes. This is a base of defining different dependability metrics used in dependability analysis of computer systems and networks. In this paper we would like to focus on more functional approach metrics which could be used by the operator of the network system. Therefore, we consider dependability of networks as a property of the networks to reliable process of user tasks, that is mean the tasks have to perform not only without faults but more with demanded performance parameters and according to the planned schedule.

We propose to concentrate the dependability analyse of the networks on fulfilling of user requirements. Therefore, it should take into consideration following aspects:

- specification of the user requirements described by task demands, for example certainty of results, confidentiality, desired time parameters etc.,
- functional and performance properties of the networks and theirs components,
- reliable properties of the network technical infrastructure that means reliable properties of the network structure and its components considered as a source of failures and faults which influence the task processing,
- process of faults management,
- threads in the network environment,
- measures and methods which are planned or build-in the network for elimination or limitation of faults, failures and attacks consequences; reconfiguration of the network is a good example of such methods,
- applied maintenance policies in the considered network.

As a consequence, a services network is considered as a dynamical structure with many streams of events generated by realized tasks, used services and resources, applied maintenance policies, manager decisions etc. Some network events are independent but other ones are direct consequences of previously history of the network life. Generally, event streams created by a real network are a mix of deterministic and stochastic streams which are strongly tied together by a network choreography. Modelling of this kind of systems is a hard problem for system designers, constructors and maintenance organizers, and for mathematicians, too. It is worth to point out some achievements in computer science area such as Service Oriented Architecture [1,3] or Business Oriented Architecture[4], and a lot of language for network description on a system choreography level, for example WS-CDL [5], or a technical infrastructure level, for example SDL [6]. These propositions are useful for analysis of a network from the designer point of view and they may been supported by simulation tools, for example modified SSF.Net simulator [7], but it is difficult to find a computer tools which are combination of language models and Monte Carlo simulators.

The paper is a step to a creation of a verbal and formal model of a net of services. It presents a generic approach to modelling performability (performance and reliability) properties of the services net. Moreover, the Petri Nets will be used for the task realization process modelling.

2. Service network – overview

We can distinguish three main elements of any network system: users, services and technical resources. As it presented in the Fig. 1 users are generating tasks which are being realized by the network system. The task to be realized requires some services presented in the system. A realization of the network service needs a defined set of technical resources. In a case when any resource component of this set is in a state "out of order" or "busy" then the network service may wait until a moment when the resource component returns to a state "available" or the service may try to create other configuration on the base of available technical resources.

Therefore, following problems should be taken into consideration:

- description and mapping a service net on existed net resources for each moment of its using;
- a prognoses process of the service net behavior in a real life conditions – definition and selection of measures;
- finding relations between measures/criteria and functional, performance and reliability parameters of the service net;
- evaluation methods of choose measures of the service net;
- decision process of maintenance organization - decision steps as a reaction on appeared events, specially on threats;

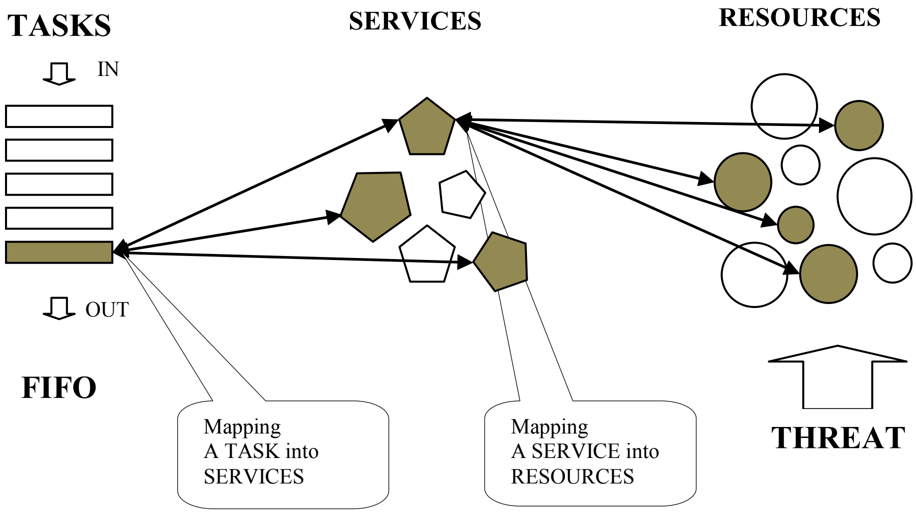
TASKS
IN
OUT
FIFO
SERVICES
RESOURCES
Mapping A TASK into SERVICES
Mapping A SERVICE into RESOURCES
THREAT

Fig. 1. Task mapping on business services and technical resources

- definition of measures and criteria of decision steps - risk of threats, and evaluation of decision risk and its cost.

3. Formal model of a service net

3.1 A service net

A functional – reliability model in the system engineering is regarded as a structured representation of the functions, activities or processes, and events generated inside of the considered system and/or by its surroundings. The system events may be divided into two main classes: functional events and reliable (together with maintenance) events. In practice this classification is very often difficult to be made because a system reaction on an event may involve a lot of functional or/and maintenance reactions. Therefore, it is better to create one common class of functional–reliable events, so called *performability* events [8]. Because these reasons considered model of services network will be called *performability* model or *functional-dependability* model [9].

The functional – dependability model of a services network has to consider specificity of the network: nodes and communication channels, the ability of dynamic changes of network traffic (routing) and reconfiguration, and all other tasks realized by the network.

The service net could be defined as a tuple:

$$SNet = \langle J, BS, TR, MS, C \rangle,$$

where:

$J = \{ J^{(i)}; i = 1,2,... \}$ – a set of tasks generated by users and realized by the service network,

$BS = \{ BS^{(b)}; b = 1,2,... \}$ – a set of services which are available in the considered network,

$TR = \{ TR^{(r)}; r = 1,2,... \}$ – technical infrastructure of the network which consists of technical resources as machines/servers, communication links etc,

MS – management system (for example - operating system),

$C = \{ c_t; t = 1,2,... \}$ – a network chronicle, defined by a set of all essential moments in a “life” of the network.

3.2 Tasks

The task $J^{(i)}$ is understood as a sequence of actions and works performed by services network in a purpose to obtain desirable results in accordance with initially predefined time schedule and data results. In this way a single task $J^{(i)} = \langle J_{IN}^{(i)}, J_{OUT}^{(i)} \rangle$ may be defined as an ordered pair of so called *input task* $J_{IN}^{(i)}$, which is described by the input parameters (postulated results and prognosis time schedule) and the corresponding *output task* $J_{OUT}^{(i)}$ (real results and real time schedule).

The input task is define as the triple:

$$J_{IN}^{(i)} = \left\langle R_P^{(i)}, A^{(i)}, C_P^{(i)} \right\rangle,$$

where

$R_P^{(i)}$- postulated results of the *i*-th task execution,

$C_P^{(i)}$- postulated chronicle of the task realization,

$A^{(i)} = A^{(i)}\left(R_P^{(i)}, C_P^{(i)}\right)$ - a sequence of actions and works necessary to obtain postulated results in planned time.

The $A^{(i)}$ may be described by a flowchart of actions and works, and its realization depends on an availability of network services and technical resources.

The output task is define as the pair

$$J_{OUT}^{(i)} = \left\langle R_{real}^{(i)},, C_{real}^{(i)} \right\rangle,$$

where:

$R_{real}^{(i)}$- real results of the *i*-th task execution,

$C_{real}^{(i)}$- real chronicle of the task realization.

The postulated results and chronicles are defined with assumed tolerance intervals ($\underline{R_P^{(i)}} \le R_P^{(i)} \le \overline{R_P^{(i)}}$ and $\underline{C_P^{(i)}} \le C_P^{(i)} \le \overline{C_P^{(i)}}$) and when the real results and chronicles are inside the intervals ($R_{real}^{(i)} \subset \left\lfloor \underline{R_P^{(i)}}, \overline{R_P^{(i)}} \right\rfloor$ and $C_{real}^{(i)} \subset \left\lfloor \underline{C_P^{(i)}}, \overline{C_P^{(i)}} \right\rfloor$) then the task is assumed to be correctly realised.

3.3 Services

The term service is understood as a discretely defined set of contiguously cooperating autonomous business or technical functionalities. Of course, a special mechanism to enable an access to one or more businesses and functionalities should be implemented in the system. The access is provided by a prescribed interface and is monitored and controlled according to constraints and policies as specified by the service description [10].

The service $BS^{(b)}$ is defined as a sequence of activities described by a set of capabilities (functionalities) $\left\{ F_k^{(b)},\ k = 1,2,... \right\}$, a set of demanded input parameters of data and/or media $BS_{IN}^{(b)}$ and a set of output parameters $BS_{OUT}^{(b)}$:

$$BS^{(b)} = \left\langle \left\{ F_k^{(b)}; k = 1,2,... \right\}, BS_{IN}^{(b)}, BS_{OUT}^{(b)} \right\rangle.$$

Because the services have to cooperate with other services than protocols and interfaces between services and/or individual activities are crucial problems which have a big impact on the definitions of the services and on processes of their execution.

A service may be realized on the base of a few separated sets of functionalities $\left\{ F_{k1}^{(b)},\ k1 = 1,2,... \right\}$, $\left\{ F_{k2}^{(b)},\ k2 = 1,2,... \right\}$ … with different costs which are the consequences of using different network resources.

3.4 Technical infrastructures

Hardware is considered as a set of hardware resources (devices and communication channels) which are described by their technical, performance, reliability and maintenance parameters. The system software is described in the same way.

3.5 Management system

The management system of service network allocates the services and network resources to realized tasks, checks the efficient states of the services network, performs suitable actions to locate faults, attacks or viruses and minimize their negative effects. Generally the management system has two main functionalities:

- monitoring of network states and controlling of services and resources,
- creating and implementing maintenance policies which ought to be adequate network reactions on concrete events/accidents. In many critical situations a team of men and the management system have to

cooperate in looking for adequate counter-measures, for instance in case of a heavy attack or a new virus.

The maintenance policy is based on two main concepts: detection of unfriendly events (attacks, faults, failures) and network responses to them. In general the network responses incorporate the following procedures:

- detection of incidents and identification of them,
- isolation of damaged network resources in order to limit proliferation of incident consequences,
- renewal of damaged services, processes and resources.

It is hard to predict all possible events (for example all new demands for a task realization) or incidents (for example failures, faults, attacks or an end of a renewal procedure) in the services network, especially it is not possible to predict all possible attacks or men faults, so system reactions are very often "improvised" by the management system, by its administrator staff or even by expert panels specially created to find a solution for the existing situation. The time, needed for the renewal, depends on the incident that has occurred, the system resources that are available and the renewal policy that is applied. The renewal policy is formulated on the basis of the required levels of system dependability and on the economical conditions (first of all, the cost of downtime and cost of lost achievements) [8,9].

Maintenance policy is based on maintenance rules that are understood as chains of decisions about allocation of services and network resources (hardware, software, information and service staff) that are undertaken to keep the system operational after an incident. These rules are very often connected with small fragments of the system, for example; replacement of a machine (a processor) or communication links. These local operations may have impact on the whole network, e.g. if a communication channel is down for a few minutes, then rates of medium (data) traffic of the network may violently change [9].

3.6 Chronicles

The set of system events is created by events connected with tasks realization, incidents occurrence (faults, viruses, and attacks) and system reactions (hardware and information renewals).

3.7 A process of the task realization

The task realization process is supported by two-level decision procedures connected with selection and allocation of the network functionalities and technical resources. There are two levels of decision process: services management and resource management. The first level of decision procedure is connected with selection suitable services and creation a task configuration. Functional and performance task demands are the base for suitable services choosing from all possible network services. The goal of the second level of the decision process is to find needed components of the network infrastructure for each service execution and the next allocate them on the base their availability to the service configuration. If any component of technical infrastructure is not ready to support the service configuration then allocation process of network infrastructure is repeated. If the management system could not create the service configuration then the service management process is started again and other task configuration may be appointed. These two decision processes are working in a loop which is started up as a reaction on network events and accidences.

On the beginning of a task realization procedure the task $J_{IN}^{(i)}$ is mapped on the network services and a subset of services $BS_s^{(i)}$ necessary for the task realization according to its postulated parameters is created; $J_{IN}^{(i)} \to BS_s^{(i)}, s = 1, 2, \ldots$. Next, a demand of technical resources for each service realization is fixed:

$BS_s^{(i)} \to R_n^{(i,s)}, n = 1,2,...$. In a real services network the same task is very often realized on the base of various service subsets and the same service may involved different technical resources. Of course, this possible diversity of task realization is connected with the flowcharts $A^{(i)}$ and the availability of network resources is checking for each service. In this way a few task configurations service configurations, additionally described by appropriately defined cost parameters, may be fund for the *i-th* task realization.

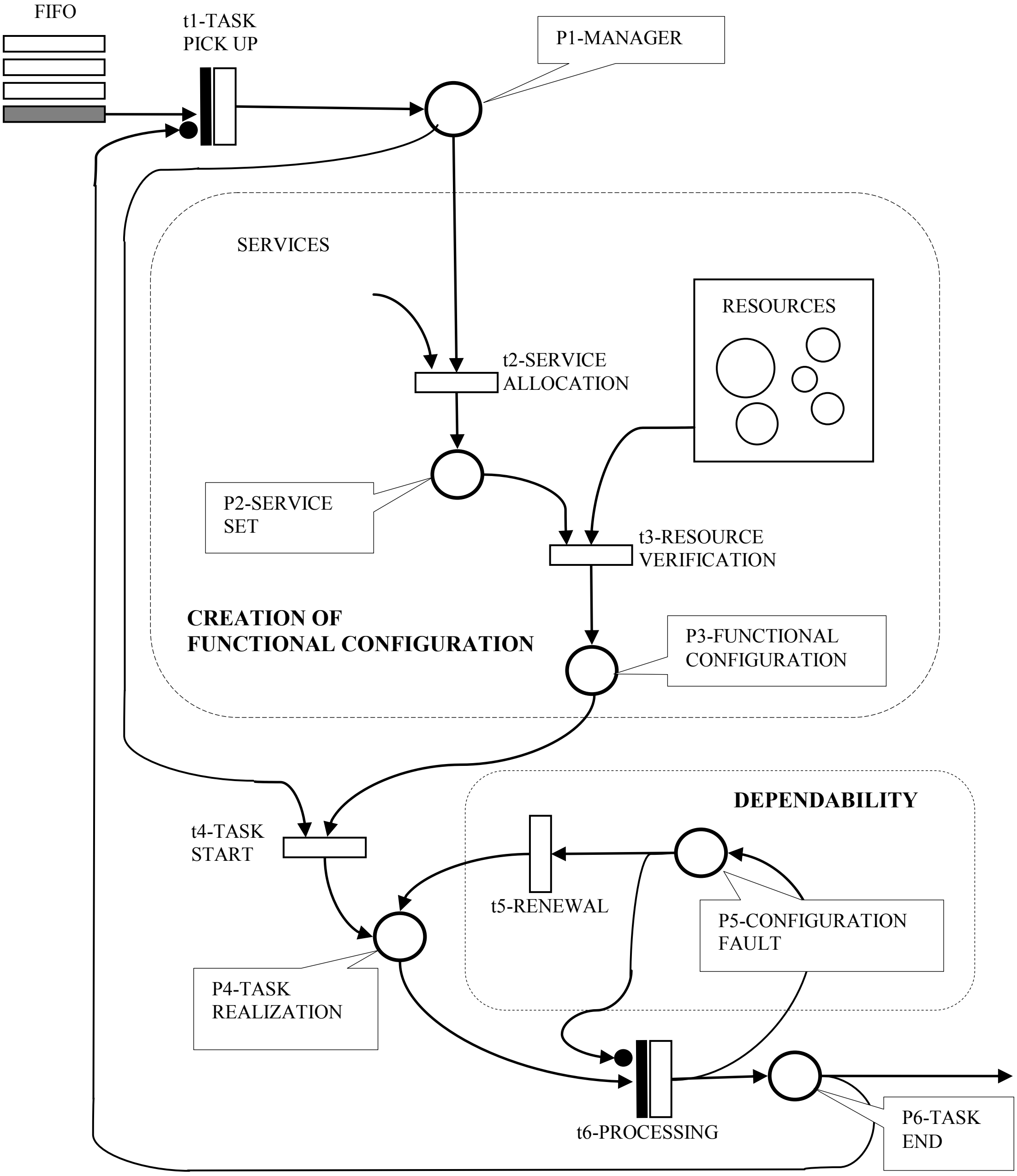

Fig. 2. The Petri net model of a task realization in a services network

4. The Petri net model

Petri Nets [12] are a powerful and often used modelling tool. They allow to represent two aspects of a modelled system static and dynamic (thanks to the token evolution). A common definition of the Petri net is formulating as a triple:

$$PN = \langle P, T, A \rangle$$

where

- P - *set of places* that represent deterministic states of processes, tasks, services, resources etc. of the considered system. The places are often complemented by tokens that are modeled abilities of these places.
- T – *set of transitions* that represent net events characterized by conditions necessary to come them into firing. The transitions are often described by firing time and other probabilistic characteristics etc.
- A – *set of arches* (*directed* and *inhibited*) that models routes on which events represented by tokens are passed by the net.

A state of the net, described by *marking* (tokens localization in the places) represents sufficient conditions for arising new events of a net's life. Net's events may be divided into many classes, for example functional, reliable or maintenance events, deterministic or probabilistic ones etc. The mention classification depends on assumed criteria.

The Petri net model of the i^{th} task realization ($J^{(i)}$) is shown on the Fig 2. It is assumed the input task ($J_{IN}^{(i)}$) is taken from the stack of waiting tasks (transition *t1* and its firing time $\tau_{T1}^{(i)}$). The choice of the task may be based on the strategy FIFO (as it is illustrated on the Fig. 2) and it is conditioned by ending of previously task (the transition *t1* is guarded by inhibited arc from the place *P6* (end of the task). The place *P1* represents the management process of mapping the input task into a set of necessary services ($BS^{(b)}$) and when the services are ready then the transition *t2* is fired (time $\tau_{T2}^{(i)}$). After checking if the chosen services may be activated on the base of needed efficient technical resources then a functional configuration of the task (place *P3*) is created (transition *t3* with time $\tau_{T3}^{(i)}$) and at this moment the manager may take a decision about start of the task process realization (transition *t4*). There is a build-in system of monitoring and detection of unfriendly accidences like faults and failures (place *P5*). When such unfriendly accidence is discovered then a renewal process of the functional configuration is started (transition *t5* and renewal time $\tau_{T5}^{(i)}$) and the task realization process is broken (the inhibited input of the transition *t6*) till the end of renewal operations.

The firing process of each transition is described by conditions (tokens in input places for the transition) which may occur with probabilities, for example a probability of a machine failure, and time duration of transition firing may be a probabilistic function, too. Of course a transition may be many times fired during a task realization, because net events may need to repeat bigger or smaller loops of the net. The Petri net model shown in the Fig. 2 is reduced and presented only to show the main idea of the proposed modelling method which may be useful for evaluation of dependability measures of services networks.

Real time of the i^{th} task realization $T_{J\,real}^{(i)}$ that is modelled as a stochastic timed Petri net with *k* transitions and *l* loops and sub loops may be evaluated as:

$$T_{J\,real}^{(i)} \cong \sum_{l \in L} \Pr\{e_l^{(i)} = 1\} \left[\sum_k \Pr\{f_{Tk,l}^{(i)} = 1\} \, \tau_{Tk,l} \right]$$

where

$e_l^{(i)} = 1$ - an event (for example, a new task, an allocation a technical resource to the i^{th} task, an end of a renewal process etc.) which is started a loop or a sub loop in the Petri net model ascribed to the i^{th} task realisation,

$f_{T\,k,l}^{(i)} = 1$ - an event; the k transition is fired during l loop connected with the i^{th} task realization.

Such dependability measures as a probability that the real time duration of the i^{th} task may be defined and evaluated on the base of the Petri net models as:

$$M_{Depend}^{(i)}(J_{IN}^{(i)}) = \Pr\left\{ T_{J\,real}^{(i)} \le T_{J\,OUT}^{(i)} \right\}.$$

5. Conclusions and further work

We have given a verbal and formal model of a service net. The formal model consists of a tuple mathematical model and the Petri Nets one. We hope that the proposed Petri net model will be very useful in the synthesis process of the service net.

Of course there are a lot problems with building the Petri net model of the real services net in which exist a large number of services and technical resources that are mapped to many concurrent realized tasks. We plan to use high level languages like for examples Business Process Modeling Notation (BPMN)[12] for a graphical representation for specifying business processes in a workflow. We hope that it could be possible to map BPMN into a Petri net model which is based on services and technical resources allocation.

Evaluation of Petri Net model and calculation of measures require a special computer tool. We plan to develop a simulation tool with a functionality similar to SOAPSimulator[13] and modified SSF.Net simulator [7] together with graphical tool for modelling and simulation.

References

[1] Gold N., Knight C. , Mohan A. and Munro M.: "Understanding service-oriented software", IEEE Software, 2004, vol. 21, pp. 71– 77.

[2] Avižienis, A., Laprie, J., Randell, B.: Fundamental Concepts of Dependability. In: 3rd Information Survivability Workshop (ISW-2000), Boston, Massachusetts, USA, 2000.

[3] Josuttis, N.: *SOA in Practice: The Art of Distributed System Design*. O'Reilly, 2007.

[4] Zhu, J. and Zhang, L. Z.: A Sandwich Model for Business Integration in BOA (Business Oriented Architecture). In Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing. APSCC. IEEE Computer Society, Washington, DC, pp. 305-310, 2006.

[5] Aime M., Atzeni A., Pomi, P., Ambra - Automated Model-Based Risk Analysis, The 3rd International Workshop on Quality of Protection Alexandria, VA, USA, 2007.

[6] Yang, H., Zhao, X., Qiu, Z., Pu, G., and Wang, S.: A Formal Model for Web Service Choreography Description Language (WS-CDL). In Proceedings of the IEEE international Conference on Web Services. ICWS. IEEE Computer Society, Washington, DC 2006.

[7] Michalska, K. and Walkowiak, T.: Hierarchical Approach to Dependability Analysis of Information Systems by Modeling and Simulation. In: Proceedings of the 2008 Second international Conference on Emerging Security information, Systems and Technologies. SECURWARE. IEEE Computer Society, Washington, DC, 356-361, 2008.

[8] Zamojski W., Caban D.; Introduction to the dependability modelling of computer systems. Dependability of Computer Systems DepCoS - RELCOMEX '06, 2006. s. 100 - 109

[9] Zamojski W., Caban D.; Maintenance policy of a network with traffic reconfiguration. Dependability of Computer Systems DepCoS - RELCOMEX '07. 2007. s. 213 – 220

[10] OASIS, Organization for the Advancement of Structured Information Standards Home Page. http://www.oasis-open.org/home/index.php

[11] Zhou, M. Kurapati V: Modeling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific Publishing, 1999

[12] White, S. A Miers, D.: BPMN Modeling and Reference Guide, 2008

[13] crosschecknet.com "SOAPSimulator" http://www.crosschecknet.com/SOAPSimulator_News_0408.php