1

Services net modeling for dependability analysis

Wojciech Zamojski and Tomasz Walkowiak
Wroclaw University of Technology
Poland

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 (Gold et al., 2004). 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 (Avižienis et al., 2000) authors described basic set of dependability attributes (i.e. availability, reliability, safety, confidentiality, integrity and maintainability). 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 the 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 (Gold et al., 2004; Josuttis, 2007) or Business Oriented Architecture(Zhu & Zhang, 2006) and a lot of languages for network description on a system choreography level, for example WS-CDL (Yang et al., 2006), or a technical infrastructure level, for example SDL (Aime et al., 2007). 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 (Zyla & Caban, 2008), but it is difficult to find a computer tools which are combination of language models and Monte Carlo (Fishman, 1996) based simulators.

The chapter presents 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. The Petri Nets will is used for the task realization process modelling. Moreover, an example of service net– the discrete transport system analysed by an event-driven simulator is presented.

2. Service network – overview

We can distinguish three main elements of any network system: users, services and technical resources. As it presented in the Figure 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:

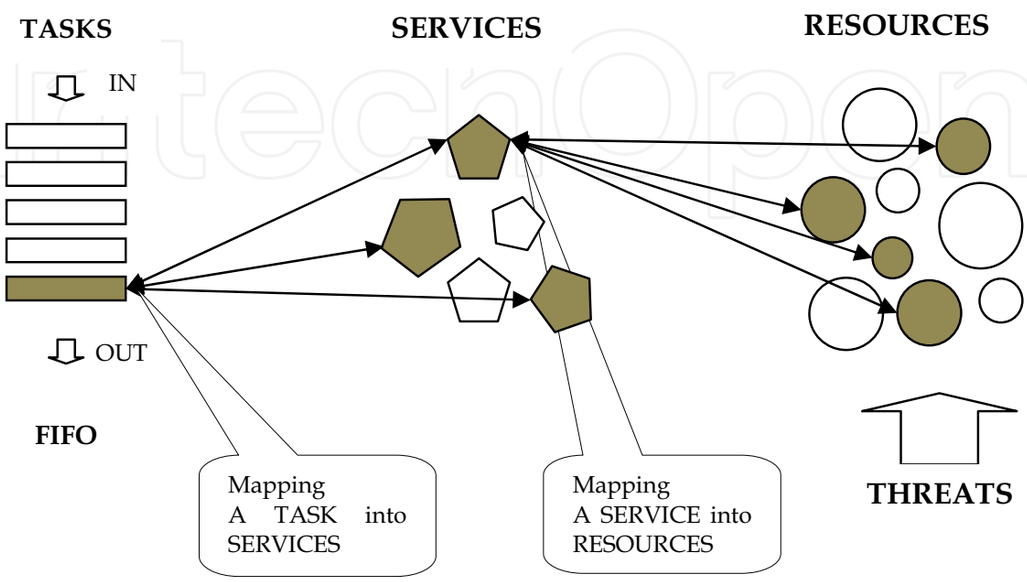


Fig. 1. Task mapping on business services and technical resources

- description and mapping a service net on existed net resources for each moment of its using;
- a prognoses process of the service net behaviour 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;
- definition of measures and criteria of decision steps - risk of threats, and evaluation of decision risk and its cost.

An illustration of problems connected with functional – dependability modelling of services networks is shown in Figure 2.

3. Functional – dependability models

The *ST model* (*State - Transition model*) is the most popular and useful methodology used in modelling of systems.
The system is considered as a union of its hardware, management system and involved personnel (administrators, users, support services etc.), so the system states depend on the states of all these elements. The system transitions are consequences of events connected

with execution of system tasks and jobs, system faults and system reactions to them, incidents, attacks and system responses etc., i.e. system events are observable occurrences which change states of the system.

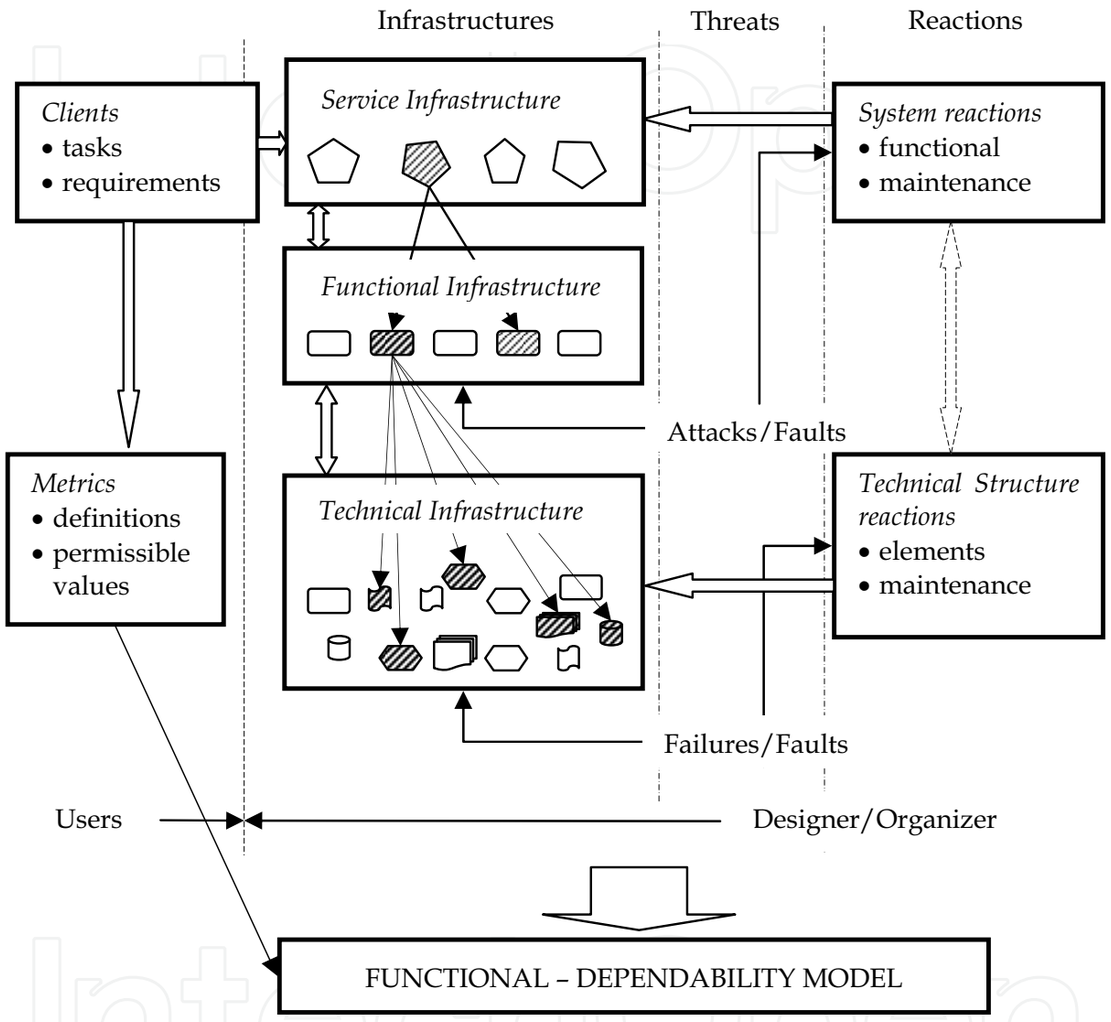


Fig. 2. Basic terms and a functional - dependability model of a services network (Zamojski, 2009)

The functional – reliability model (Zamojski, 2005) of computer system S_C is a configuration of hardware H, software SP, men M, management system (operating system) MS, tasks (functions) J and system events E_S

$$S_C \subset H \times SP \times J \times M \times MS \times E_S \tag{1}$$

The system events includes those connected with tasks realization, occurrence of incidents (faults, viruses, and attacks) and system reactions to them (hardware and information

renewals). The system events are very often described by their time parameters which are collected in so called *a chronicle* of the system.

A functional configuration $S_C^{(i)}$ of the computer system is a set of hardware and software resources that are allocated to realize *i*-th task $j^{(i)}$;

$$\left(j^{(i)} \subset J\right) \Rightarrow \left(S_C^{(i)} \subset S_C\right) \tag{2}$$

and

$$S_C^{(i)} \subseteq H^{(i)} \times SP^{(i)} \times j^{(i)} \times M^{(i)} \times MS^{(i)} \times E_S^{(i)} \tag{3}$$

where superscript *(i)* fix subsets of system resources needed for execution *i*-th task.

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 (Zamojski & Caban, 2006). Because of these reasons considered model of services network will be called *performability* model or *functional-dependability* model (Zamojski & Caban, 2007).

If the functional – reliability model is built as the ST model then the set of the system states is determined by the states of all resources involved in tasks realized at the moment. The system resource allocations are dynamic, modified due to the incoming tasks, occurring incidents and system reactions (especially reconfiguration).

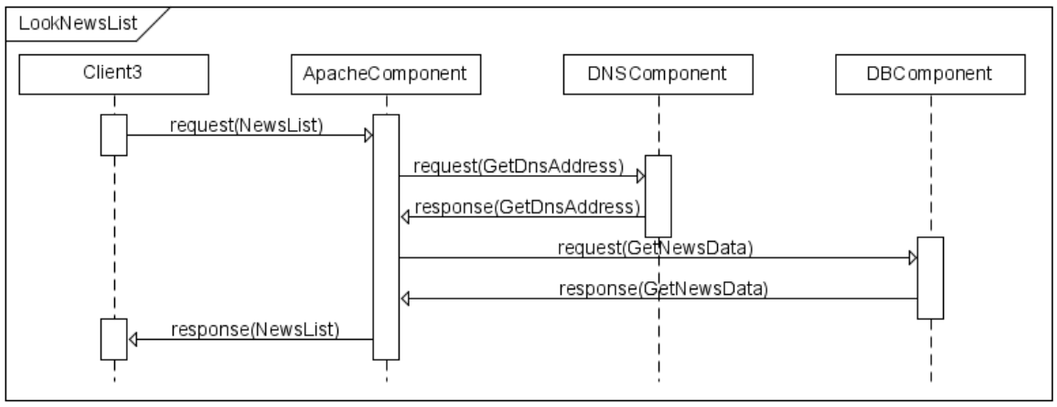


Fig. 3. Exemplar choreography

4. Formal model of a service net

4.1 A service net

A *services net* is a system of business services that are necessary for user (clients) tasks realization process. The services net are built on the bases of technical infrastructure

(*technological resources*) and *technological services* which are involved into a task realization process according to decisions of a management system. The task realization process may include many sequences of services, functions and operations which are using assignment network resources - in the computer science this process of assignments and realization steps is called as *a choreography*. An example of choreography for web service is presented in Figure 3.
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 network could be defined as a tuple:

$$SNet = \langle J, BS, TR, MS, C \rangle, \tag{4}$$

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.

4.2 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:

$$J_{IN}^{(i)} = \langle R_P^{(i)}, A^{(i)}, C_P^{(i)} \rangle, \tag{5}$$

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 $\mathbf{A^{(i)}}$ 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:

$$J_{OUT}^{(i)} = \left\langle R_{real}^{(i)},, C_{real}^{(i)} \right\rangle, \tag{6}$$

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 ($R_{real}^{(i)} \subset \left[\underline{R_P^{(i)}}, \overline{R_P^{(i)}} \right]$ and $C_{real}^{(i)} \subset \left[\underline{C_P^{(i)}}, \overline{C_P^{(i)}} \right]$) then the task is assumed to be correctly realised.

4.3 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[1].
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 $BS_{OUT}^{(b)}$:

$$BS^{(b)} = \left\langle \left\{ F_k^{(b)}; k = 1,2,... \right\}, BS_{IN}^{(b)}, BS_{OUT}^{(b)} \right\rangle. \tag{7}$$

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.

4.4 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.

---

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

4.5 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) (Zamojski & Caban, 2006; Zamojski & Caban, 2007).

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 (Zamojski & Caban, 2007).

4.6 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).

4.7 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; $J_{IN}^{(i)} \rightarrow BS_s^{(i)}$. Next, a demand of technical resources for each service realization is fixed: $BS_s^{(i)} \rightarrow R_n^{(i,s)}$. 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.

5. The Petri net model

Petri Nets (Zhou & Kurapati, 1999) 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:

$$PN = \langle P, T, A \rangle \tag{8}$$

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.

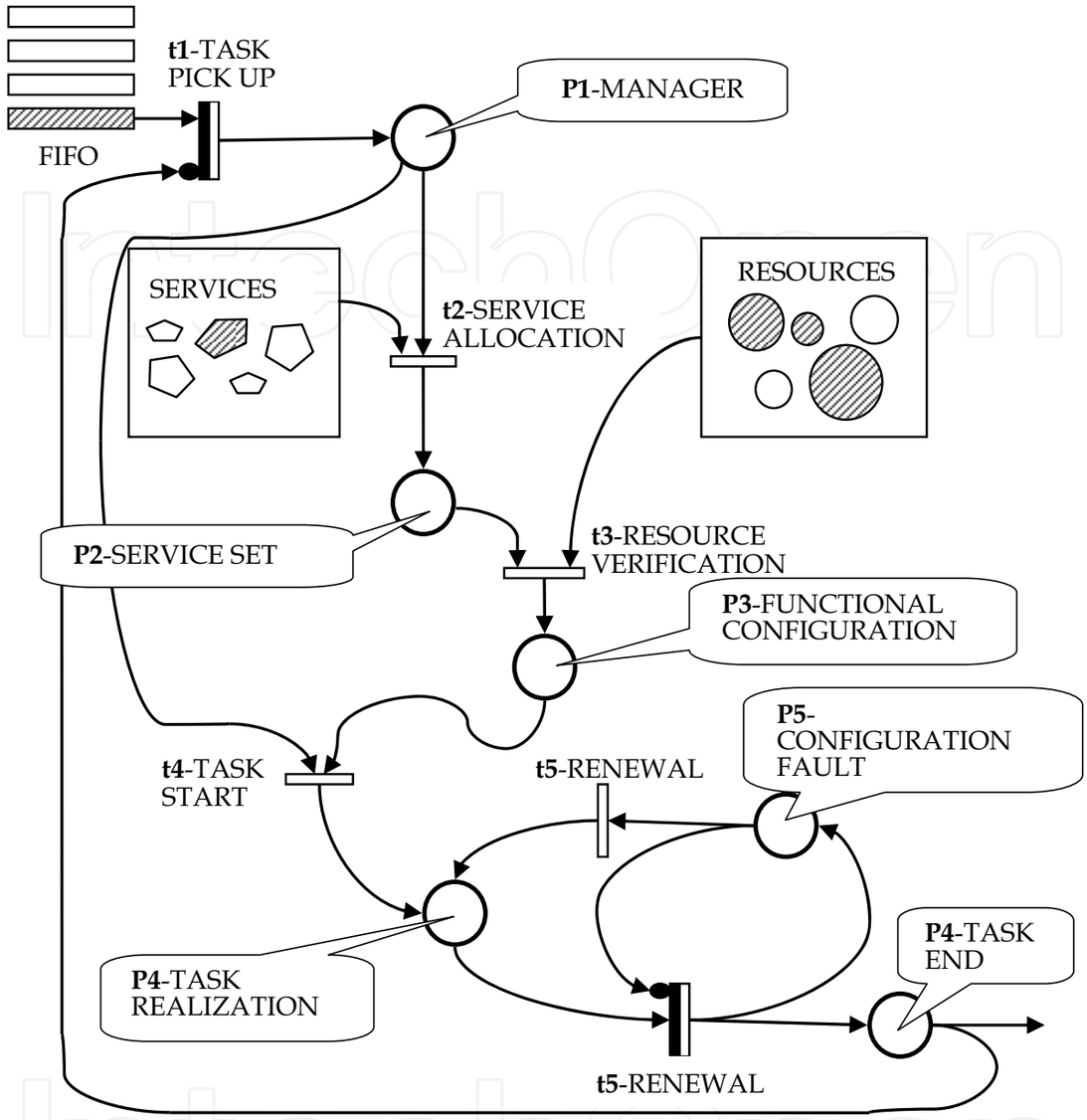


Fig. 4. The Petri net model of a task realization in a services network

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 in the Figure 4. 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 Figure 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 Figure 4 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:

$$T_{J\,real}^{(i)} \cong \sum_{l \in L} \Pr\left\{e_l^{(i)} = 1\right\} \left[\sum_{k} \Pr\left\{f_{Tk,l}^{(i)} = 1\right\} \tau_{Tk,l} \right], \tag{9}$$

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 ith task realisation,
- $f_{Tk,l}^{(i)} = 1$ - 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:

$$M_{Depend}^{(i)}(J_{IN}^{(i)}) = \Pr\left\{T_{J\,real}^{(i)} \le T_{J\,OUT}^{(i)}\right\}. \tag{10}$$

6. Discrete transport system – service net case study

An example of service net could be a DTSCNTT - Discrete Transport System with Central Node and Time-Table (Walkowiak et al., 2007). This is a simplified case of the Polish Post transport system.
Following the definition (4) each elements of service net could be described as follows.

The business service (*BS*) provided the Polish Post and therefore DTSNTT service net is the delivery of mails. The technical infrastructure (*TR*) consists of a set of nodes placed in different geographical locations and set of vehicles and timetable. There are bidirectional routes between nodes marked by lines. There is distinguished one node called central mode. Mails are distributed among nodes by vehicles.

Each vehicle is described by following functional and reliability parameters: mean speed of a journey, capacity – number of containers which can be loaded, reliability function and time of vehicle maintenance.

Management system (*MS*) is defined by time table since vehicles distributing mails among system nodes operate according to the time-table exactly as city buses or intercity coaches. The time-table consists of a set of routes (sequence of nodes starting and ending in the central node, time of approaching each node in the route and the recommended size of a vehicle). The number of used vehicle, or the capacity of vehicles does not depend on temporary situation described by number of transportation tasks or by the task amount for example. It means that it is possible to realize the journey by completely empty vehicle or the vehicle cannot load the available amount of commodity (the vehicle is to small). Time-table is a fixed element of the system in observable time horizon, but it is possible to use different time-tables for different seasons or months of the year.

To reduce the complexity of the model we have decided to model the containers not separate mails (Walkowiak & Mazurkiewicz, 2009). Therefore, the tasks (*J*) of sending mails is modelled as a random process of containers generation. Each generated container has a destination address. The central node is the destination address for all containers generated in the ordinary nodes. Where containers addressed to in any ordinary nodes are generated in the central node. The generation of containers is described by Poisson process. In case of central node there are separate processes for each ordinary node. Whereas, for ordinary nodes there is one process, since commodities are transported from ordinary nodes to the central node or in opposite direction. Postulated result of any task is to transport a container to the destination node within a given time limit.

The process of any task realization could be described as follows. The container is generated in some node at a given time (according to Poisson process) and stored in the node waiting for the vehicle to be transported to the destination node. Each day a given time-table is realized, it means that at a time given by the time table a vehicle, selected randomly from vehicles available in the central node, starts from central node and is loaded with containers addressed to each ordinary nodes included in a given route. The loading is done in a service point. This is done in a proportional way. Since the number of service points is limited (parameter of the central node) and loading takes some time is there is no free service point vehicles has to wait in a queue. After loading the vehicle goes to a given ordinary node - it takes some time according to vehicle speed - random process and road length. After approaching the ordinary node the vehicle is waiting in an input queue if there is any other vehicle being loaded/unloaded at the same time. The containers addressed to given node are unloaded and empty space in the vehicle is filled by containers addressed to a central node. The operation is repeated in each node on the route and finally the vehicle is approaching the central node when is fully unloaded and after it is available for the next route. The process of vehicle operation could be stopped at any moment due to a failure (described by a random process). After the failure, the vehicle waits for a maintenance crew

(if it is not available due to repairing other vehicles), is being repaired (random time) and after it continues its journey (Walkowiak & Mazurkiewicz, 2009).
As suggested in the introduction the simulator tool for analysing DTSCNTT service net was developed. The tool was adopting the event simulation approach, which is based on a idea of event, which could be described by time of event occurring, type of event (in case of DTSCNTT it could be a vehicle failure) and element or set of elements of the system on which event has its influence. The simulation is done by analyzing a queue of event (sorted by time of event occurring) while updating the states of system elements according to rules related to a proper type of an event. (Walkowiak et al., 2007)
We proposed for the case study analysis an exemplar DTSCNTT based on Polish Post regional centre in Wroclaw. We have modelled a system consisting of one central node (Wroclaw regional centre) and twenty two other nodes - cities where there are local post distribution points in Dolny Slask Province. The length of roads were set according to real road distances between cities used in the analyzed case study. The intensity of generation of containers for all destinations were set to 4,16 per hour in each direction giving in average 4400 containers to be transported each day. The vehicles speed was modelled by Gaussian distribution with 50 km/h of mean value and 5 km/h of standard deviation. The average loading time was equal to 5 minutes. There were two types of vehicles: with capacity of 10 and 15 containers. The MTTF of each vehicle was set to 2000. The average repair time was set to 5h (Gaussian distribution). (Walkowiak & Mazurkiewicz, 2009)
The simulation time was set to 100 days and each simulation was repeated 10.000 times. We have calculated the dependability measure defined by (10), the probability that the duration time of a task (delivery of some container) will be longer then a given time limit using Monte-Carlo approach (Fishman, 1996). The achieved results are presented in Figure 5.

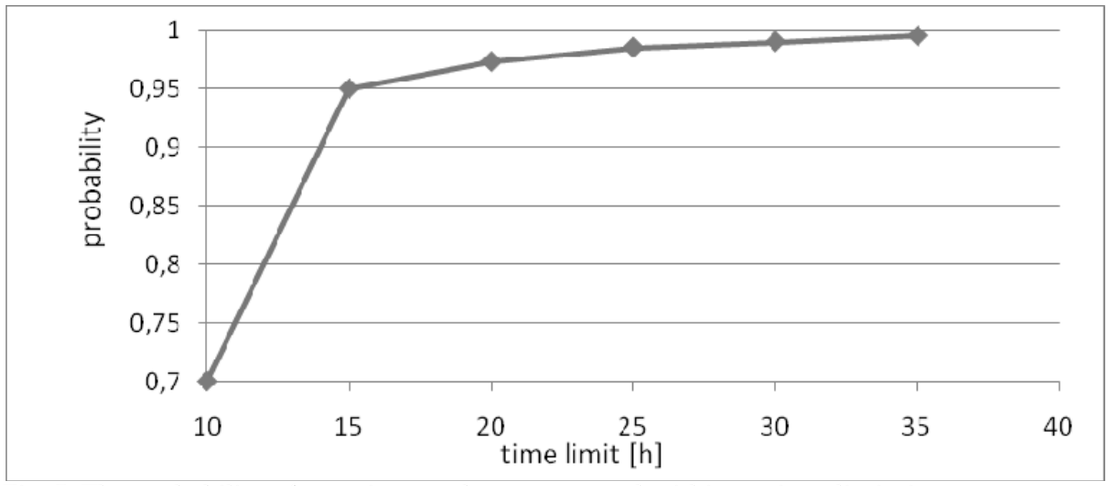


Fig. 5. The probability of containers to be transported within a given limit time

7. Conclusion

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 have also presented an exemplar case study of service net a discrete transport system service net – a simplified case of Polish Post transport system. It was analysed by a usage of a discrete transport system simulator.
We plan to develop a simulation tool for a generic service nets with a functionality similar to presented discrete transport system simulator or BS.SSF simulator (Walkowiak, 2009) together with graphical tool for modelling and simulation. We also plan to use high level languages like for examples Business Process Modeling Notation (White & Miers 2008) 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 or a general purpose service net simulator allowing to perform a service net dependability analysis.

8. References

Aime, M.; Atzeni, A.; Pomi, P. (2007). Ambra - Automated Model-Based Risk Analysis, *Proceedings of the 3rd International Workshop on Quality of Protection,* pp. 43-48, Alexandria, ACM, New York

Avižienis, A. ; Laprie, J. ; Randell, B. (2000). Fundamental Concepts of Dependability. *Proceedinggs of 3rd Information Survivability Workshop*, Boston

Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York

Gold, N.; Knight, C. ; Mohan, A.; Munro, M. (2004). Understanding service-oriented software. *IEEE Software*, Vol. 21, 71– 77

Josuttis, N. (2007). *SOA in Practice: The Art of Distributed System Design*, O'Reilly

Walkowiak, T. (2009). Information systems performance analysis using task-level simulator, *Proceedings of International Conference on Dependability of Computer Systems*, pp. 218-225, Brunow, IEEE Computer Society Press, Los Alamitos

Walkowiak T. ; Mazurkieiwicz, J. (2009), Analysis of critical situations in discrete transport systems, *Proceedings of International Conference on Dependability of Computer Systems*, pp. 364-371, Brunow, IEEE Computer Society Press, Los Alamitos

Walkowiak, T. ; Mazurkiewicz, J.; Kaplon, K. (2007). Functional analysis of discrete transport system realized by SSF simulation tool. *Advances simulation of systems. Proceedings of the XXIXth International Autumn Colloquium*, pp. 103-108, Sv. Hostyn, MARQ, Ostrava

White, S. A Miers, D. (2008). *BPMN Modeling and Reference Guide*, Future Strategies Inc., Lighthouse Pt

Yang, H.; Zhao, X.; Qiu, Z.; Pu, G; Wang, S. (2006). A Formal Model for Web Service Choreography Description Language (WS-CDL). *Proceedings of the IEEE international Conference on Web Services*, IEEE Computer Society, Washington

Zamojski, W. (2005). Functional-reliability model of computer-human system. *Computer engineering*, pp. 278-297, Eds. Wojciech Zamojski, WKL, Warszawa (*in Polish*)

Zamojski, W. (2009). Dependability of services networks. *Proceedings of the Third Summer Safety and Reliability Seminars*, pp. 387-396, Gdnask-Sopot, Polish Safety and Reliability Association, Gdansk

Zamojski W.; Caban D. (2006). Introduction to the dependability modelling of computer systems. *Proceedings of International Conference on Dependability of Computer Systems,* pp. 100 – 109, Szklarska Poreba, IEEE Computer Society Press, Los Alamitos

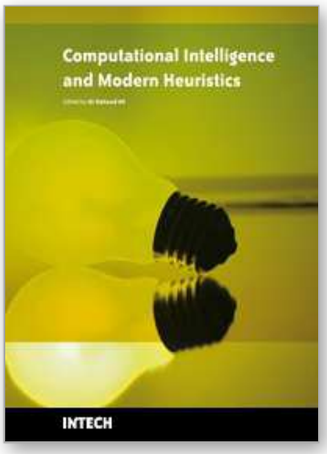
Zamojski, W.; Caban, D. (2007). Maintenance policy of a network with traffic reconfiguration. *Proceedings of International Conference on Dependability of Computer Systems*, pp. 213 – 220, Szklarska Poreba, IEEE Computer Society Press, Los Alamitos

Zhu, J.; Zhang, L. Z. (2006). A Sandwich Model for Business Integration in BOA (Business Oriented Architecture). *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing*, pp. 305-310, IEEE Computer Society, Washington

Zhou, M.; Kurapati, V (1999) *Modeling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach*. World Scientific Publishing

Zyla, M.; Caban, D. (2008). Dependability Analysis of SOA systems. *Proceedings of International Conference on Dependability of Computer Systems*, pp. 301–306, Szklarska Poreba, IEEE Computer Society Press, Los Alamitos

IntechOpen



Computational Intelligence and Modern Heuristics

Edited by Al-Dahoud Ali

ISBN 978-953-7619-28-2
Hard cover, 348 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

The chapters of this book are collected mainly from the best selected papers that have been published in the 4th International conference on Information Technology ICIT 2009, that has been held in Al-Zaytoonah University, Jordan in the period 3-5/6/2009. The other chapters have been collected as related works to the topics of the book.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen