Transport and Telecommunication, 2013, volume 14, no. 4, 262–271
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2013-0022

NETWORK SYSTEMS ANALYSIS IN CASE OF CRITICAL SITUATIONS

Jacek Mazurkiewicz

Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology
ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
E-mail: Jacek.Mazurkiewicz@pwr.wroc.pl

The paper describes the analysis and discussion of the network systems in case of the critical situation that happens during ordinary work. The formal model is proposed – based on the two types of real sophisticated network systems – with the approach to its modelling based on the system behaviour observation. The definition of the critical situation sets are created by reliability, functional and human reasons. No restriction on the system structure and on a kind of distribution describing the system functional and reliability parameters is the main advantage of the approach. The proposed solution seems to be essential for the owner and administrator of the transportation systems.

Keywords: network systems, critical sets, reliability, dependability modelling

1. Introduction

Contemporary network systems are very often considered as a set of services realised in well-defined environment created by the necessary hardware and software utensils. The system's dependability can be described by such attributes as *availability* (readiness for correct service), *reliability* (continuity of correct service), *safety* (absence of catastrophic consequences on the users and the environment), *security* (availability of the system only for authorized users), *confidentiality* (absence of unauthorized disclosure of information), *integrity* (absence of improper system's state alterations) and *maintainability* (ability to undergo repairs and modifications) [1, 3, 8, 13].

The system realises some tasks and it is assumed that the main system goal, taken into consideration during design and operation, is to fulfil the user requirements. The system's functionalities (services) and the technical resources are engaged for task realisation. Each task needs a fixed list of services which are processed based on the system's technological infrastructure. The different services may be realised using the same technical resources and the same services may be realised involving different sets of the technical resources. It is easy to understand that the different values of performance and reliability parameters should be taken into account. The last statement is essential when tasks are realised in the real system surrounded by unfriendly environment that may be a source of threads and even intentional attacks.

Moreover, the real systems are built on the base of unreliable technical infrastructures and components. The modern systems are equipped with suitable measures and probes, which minimise the negative effects of these inefficiencies (a check-diagnostic complex, fault recovery, information renewal, time and hardware redundancy, reconfiguration or graceful degradation, restart etc). The contemporary network systems are created as very sophisticated products of human idea characterized by the complex structure. This way the critical situations observable during its exploitation are not always predictable for system's owners and managers, but could be very costly for a company and sometimes even damaging.

The aim of this paper is to point the problems of the critical situations in unified network system – product of essential elements and features taken from two kind of real systems: ***Discrete Transport System (DTS)*** and ***Computer Information System (CIS)***. Each part of the system is characterised by unique set of features and can caused the critical situation of whole system if it starts to work in unusual way or the fault or error of it is noticed. It is hard for an administrator, manager or an owner to understand the system's behaviour and to combine the large scale of variant states of it in single – easily observable and controlled global metric as a pointer to make the proper decision in short time period. To overcome this problem we propose a functional approach. The system is analysed from the functional point of view, focusing on business service realized by a system [14]. The analysis is following a classical [15]:

modelling and simulation approach. It allows calculating different system's measures, which could be a base for decisions related to administration of the transportation systems.

The results of the system observation – understand as the set of data collected during the simulation process are the basis to define the critical situations and they allow providing proper solution to lift-up the systems in effective way if the critical situation occurs. This is the only sensible way, because the critical situations are the real and not removable part of the system life. The organization of this paper is as follow. We start with description of the abstract service network model (Section 2). Base in it we define the normal conditions of the system's work (Section 3). In Section 4 we provide the most adequate – in case of the level of detail – the well-established description of the critical situation.

2. Service Network Model - Idea

The paper describes approach based on functional-dependability models understood as a concept of specifying dependability aspect for two perspectives: secure and dependable system as much as service-related operational system. In our research, we focus on two types of service models, that where close to our interest area: ***Discrete Transport System*** (***DTS***) [13, 14, 15, 16, 17] and ***Computer Information System*** (***CIS***) [8, 10, 11, 12]. Both systems can be analysed separately, but because of their specific goal, some common mechanisms can be seen. Taking into consideration more generic perspective, we decided to focus on a common view on the system model we call - ***Abstract Service Network Model***.

As mentioned, both systems have the same aim – to provide a service in a sense of user request accomplishment. For this reasons, the key point to analyse the systems is a *Task* (*T*) given to the systems. Task is defined by the user and parameters related with time (user patience time, delivery take, etc.) but also it is strongly and inextricably connected with some service scenario. In fact, when we analyse logically the way the service is provided, we can see that the scenario conditions define specific choreography (graph of various components) within a service.

The choreography must be defined and known. Since task is realized as an input to the *Business Service* (*BS*), therefore its choreography is based on predefined service components located in network nodes (reconfigurable components). Moreover, network nodes base on *Technical Infrastructure* (*TI*) – resources used as elements for providing dependable service seen as a hardware and software linked within a network. Various functional define each element of the Technical Infrastructure (routes and central points in *Discrete Transport System*, computers or network devices in *Complex Information Systems*) and dependability parameters, not to mention about some time functions. Time related with the technical resources is as much important as time on a service level, therefore we speak about – *Chronicle of the System* (*K*).

Taking into consideration these common features an abstract model can be proposed as follows:

$$ANS =< T, BS, TI, M, K >, \tag{1}$$

where

ANS – Abstract Network Services,
T – Task,
BS – Business Service,
TI – Technical Infrastructure,
M – User,
K – Chronicle of the System.

The unified description can guarantee the required level of abstraction for the analysis we are going to provide.

3. Network Service Description

3.1. Tasks Problem

The problems of the contemporary systems reliability certainly need to be extended to cover the envisaged fact that the main object (system) of its studies is a tightly connected complex of hardware resources, information resources (algorithms and procedures of operations and system management) and human-factor (managers, administrators and users).

The studied systems realize complex functions and are capable of substituting tasks on detecting faults (functional redundancy). The systems operate in a changing environment, often antagonistic to them. Users generate tasks which are being realised by the system. The task to be realised requires some services (functionalities) available in the system. A realisation of the 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", the task may wait until a moment when the resource component returns to a state "available" or the service may try to create other configuration based on available technical infrastructure [2, 3, 4].

A technological infrastructure is considered as a set of hardware resources (devices and communication channels) which are described by sets of their technological, reliability and maintenance parameters. The information resources are understood in the same way. The human-factor's functions are defined little bit different: she or he can be defined as: a system operator, a service person, a system manager (administrator), etc. [21, 22]. The system management allocates the resources to the task realisation, checks the efficient states of the system, performs the suitable actions to locate faults, attacks or viruses and to minimise their negative effects. In many situations the system staff and the management system have to cooperate in looking for adequate decisions (for instance, to fight with a heavy attack or when a new virus is disclosed). The system events corresponds to: tasks realisation, occurrence of incidents (faults, viruses, attacks) and system's reactions to them (technological and information renewals). Task configurations change when the tasks are being processed. The software management, reacting with the system's users, determines the changes. Some changes may be the result of detecting system's faults and reacting to them. This is called system's reconfiguration [18, 20]. The subsets of resources used by the tasks do not need to be disjoint. A resource that can be allocated to more than one configuration at the same time is called sharable, whereas one that cannot is non-sharable. Some resources, for example the central processors in computer systems, are "time-sharable". This is a technique that allows sharing of resources that are essentially non-sharable, by very fast switching of the allocation in time [1, 10, 11].

3.2. Events

Different events of the service network are considered as:

- "normal" functional events described by such time parameters as the start or / and the end of the task, a moment of a system resources allocation, a time of occurrence of a new task, an (prognoses or real) task execution time, etc.,
- unfriendly incidents that are disturbed efficient system's execution; for example failures of transport structure, failures and errors, delay time of data packages, faults of network devices or dispatching system, etc.

It is easy to notice that the first class of system events is strictly connected with correct system's task realization and the second one groups all events disrupting the efficient operation of the system and which may start the system's defence reactions. In this way the first class of events will be called "efficient functional events" and the second one "dependable incidents" or "unfriendly events". A classification of dependable incidents and system's reactions is presented in the Figure 1.

A dependable incident is an event that might lead to some disruptions in the system's behaviour. The incident may cause some damage to the system's resources; transport structure, management actions and, in consequence, it may disrupt the executed transport processes [3, 4].

If a fault appears during the task execution then the system on the base of decision of its management system starts renewal processes. Time of technological renewal activities are added to the nominal time of the task so a real time of the task duration will be longer. The real duration time of the executed tasks depends on the nature of the system's faults. Failures of hardware may need both renewals of technological resources and information resources. Consequences of human errors or computer software faults are limited to renewals of information processes. Sometimes an incident, which has occurred in a short time interval, may have a more serious impact on the system's behaviour; it may escalate to a security incident, a crisis or a catastrophe. The failures of the network structure - physical failures of technical infrastructure need to use adequate service teams, spare elements or substituted routes. Very often "technical" system's renewal processes are considered with assuming of the limited resources, for example the number service team for the part of the network [5, 20].

Other sources of the network disruptions we can find in organization and management:

- overloading of the technical infrastructure,
- traffic problems or jams – caused by limited bandwidth or dispatching errors,

- dispatching faults – system is not able to keep up the dynamic changes of the situation in the working network.

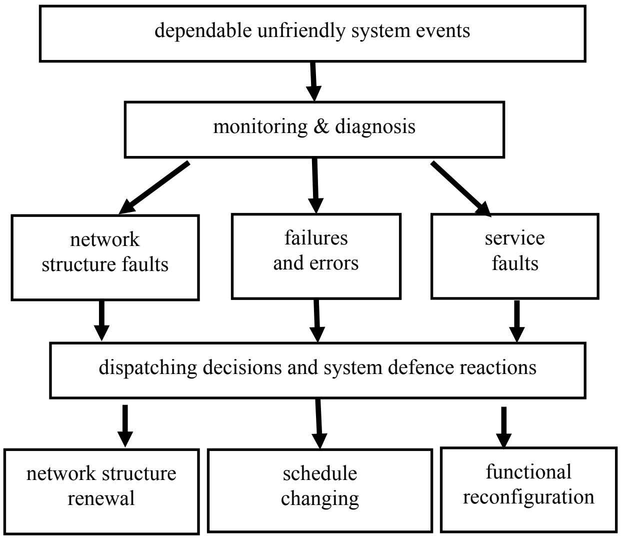
dependable unfriendly system events
monitoring & diagnosis
network structure faults
failures and errors
service faults
dispatching decisions and system defence reactions
network structure renewal
schedule changing
functional reconfiguration

Figure 1. A classification of unfriendly events of a discrete transport system

In these cases exploitation system's renewal processes are initiated by the system's dispatcher. The processes very often consume more time and money than a renewal of a "simple (physical)" broken technical resource, e.g. a repair of a failed truck or a lift.

3.3. Maintenance

The modern systems are equipped with suitable measures, which minimise the negative effects of these inefficiencies (a check-diagnostic complex, fault recovery, information renewal, time and hardware redundancy, reconfiguration or graceful degradation, restart etc). The special services resources (service persons, different redundancy devices, etc.) supported by the so-called maintenance policies (procedures of the service resources using in purpose to minimise negative consequences of faults that are prepared before or created ad hoc by the system's manager) are build in every real system [3, 4, 17, 20]. The maintenance policy is based on two main concepts: detection of unfriendly events and system's responses to them. Detection mechanisms should ensure detection of incidents based on observation of a combination of seemingly unrelated events, or on an abnormal behaviour of the system. Response provides a framework for counter-measure initiatives to respond in a quick and appropriate way to detected incidents.

In general, the system's responses incorporate the following procedures:

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

Relations among the incidents and the reactions of the system are shown on Figure 2.

A services network is a system of functional services that are necessary for clients' tasks realisation process. The services networks are organized based on the technical infrastructure and technological services which are involved into a task realisation process according to decisions of the management system. The task realisation process may include many sequences of services, functions and operations which are using assignment network resources. Description of the allocation of network services and their implementation process will be hereinafter referred to as network choreography. We can build more general definition of the system introducing the idea of the net of services. It is described at the upper level of abstraction: a task or a job may use a single service or a few services – concurrent or sequenced – on the base of available network resources.

The management system – allocates services (functionalities) and network resources to realized tasks, checks states of the services network and controls suitable system's responses to detected and localized unfriendly events and minimizes their negative effects. The control of the defense reactions of the system is understood as the choice of an appropriate maintenance policy.

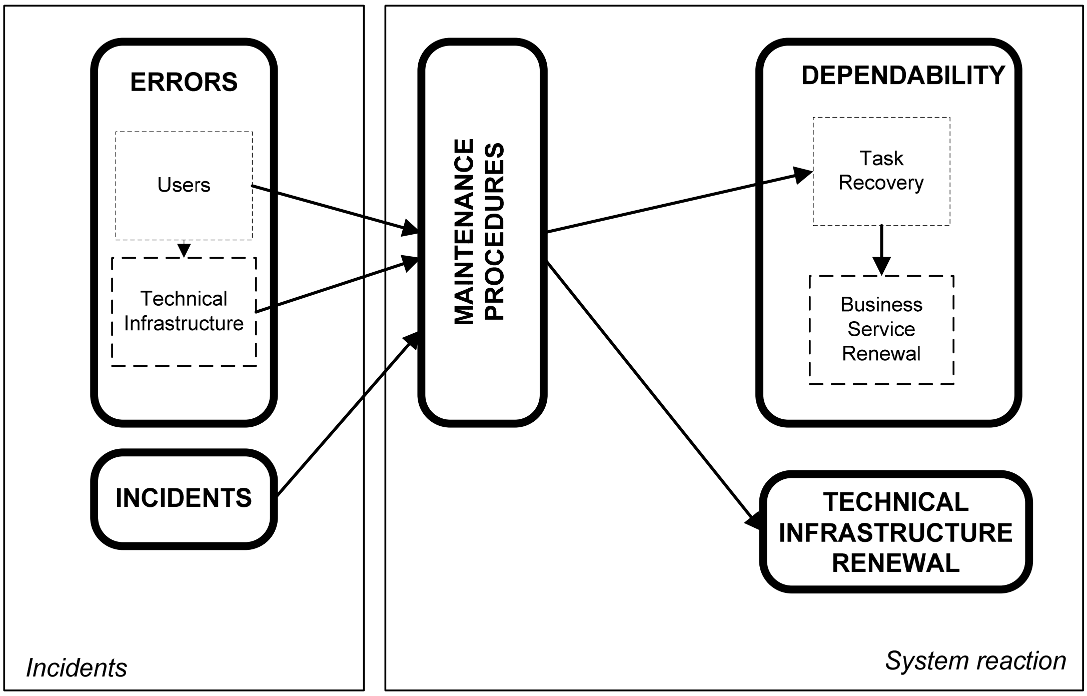

Figure 2. Incidents and reactions of the system

A service may be realised based on a few separated sets of functionalities with different costs which are the consequences of using different network resources. 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.

Generally the management system has 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 persons and the management system have to cooperate in looking for adequate counter-measures.

As a consequence, the 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 may be independent but majority of events depends on a history of a network life. Generally, event streams created by a real network are a mix of deterministic and stochastic streams which are strongly tied together by network choreography. Modelling of this kind of systems is a hard problem for system's designers, constructors and maintenance organizers, and for mathematicians, too.

It is proposed to focus the dependability analysis of the networks on the fulfilment of requirements defined by user task [20]. Therefore, it should take into consideration following aspects:

- specification of the user requirements described by task demands, for example expected volume to transport, desired time parameters,
- functional and performance properties of the network system and theirs components,
- reliable properties of the technical infrastructure that means reliable properties of the network structure and its components considered as a source of unfriendly events which influence the task processing,
- threads in the network environment,
- measures and methods which are planned or build-in the network system for elimination or limitation of unfriendly incident consequences; reconfiguration of the transport system is a good example of such methods,
- the system of maintenance policies applied in the considered network.

The task realisation process is supported by two-level decision procedures connected with selection and allocation of the network services (functionalities) and infrastructure resources. The first level of decision procedure is focused on suitable services selection and a task configuration. The functional and the performance task demands are based on 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 to allocate them based on their availability to the service configuration. If any component of technical infrastructure is not ready to support the service configuration then the 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 incidents [3, 4, 5, 20].

3.4. Dependability Problems

The term dependability is well known in the literature and commonly used by fault tolerant and dependable computing community, but has been assigned many different meanings. For example, there is more than one definition of dependability [3, 4, 6, 7, 9].

The dependability of the system can be defined as the ability to execute the functions (tasks, jobs) correctly, in the anticipated time, in the assumed work conditions, and in the presence of threats, technological resources failures, information resources and human faults (mainly malfunctions) [5]. Dependability is the most comprehensive concept for modelling complex systems taking a top-down approach [1].

It is evolving into a distinct discipline attempting to subsume the preceding concepts of reliability, and fault-tolerance. There is no universally accepted definition of dependability; the term has been accepted for use in a generic sense as an umbrella concept [2, 3].

Users of the system realise some tasks using it – for example: send a parcel in the transport system or buy a ticket in the internet ticket office. It is assumed that the main goal, taken into consideration during design and operation, is to fulfil the user’s requirements. We can easy find some quantitative and qualitative parameters of user’s tasks [2, 20].

The system’s functionalities (services) and the technical resources are engaged for task realisation. Each task needs a fixed list of services, which are processed based on the system’s technological infrastructure or the part of it. The different services may be realised using the same technical resources and the same services may be realised involving different sets of the technical resources. It is easy to understand that the different values of performance and reliability parameters are taking into account.

The last statement is essential when tasks are realised in the real system surrounded by unfriendly environment that may be a source of threads and even intentional attacks. Moreover, the real systems are built of unreliable software and hardware components as well.

Therefore, it should take into consideration following aspects:

- specification of the user requirements described by task demands,
- functional and performance properties of the system and theirs components,
- reliable properties of the system’s technological infrastructure that means reliable properties of the system’s structure and its components considered as a source of failures and faults which influence the task processing,
- process of faults management,
- threads in the system’s environment,
- measures and methods which are planned or build-in to eliminate or reduce the faults, failures and attacks consequences,
- applied maintenance policies (together with their costs) in the considered system.

It is hard to predict all incidents in the system; especially, it is not possible to envision all possible attacks, so system’s reactions are very often "improvised" by the system, by the 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’s resources that are available and the renewal policy that is applied. The renewal policy should be formulated on the basis of the required levels of system’s dependability (and safety) and on the economic conditions (first of all, the cost of downtime and lost processing/computations) [2].

As a consequence, a system is considered as a dynamic structure with many streams of events generated by realised tasks, used services and resources, applied maintenance policies, manager decisions, etc. Some network events are independent but other can be found as 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 network choreography. Modelling of this kind of systems is a hard problem for system's designers, constructors and maintenance organisers, as well as for mathematicians. It is worth to point out some achievements in the computer science area such as Service Oriented Architecture [3, 4, 19] or Business Oriented Architecture [19, 22], and a lot of languages for network description on a system's choreography level, for example *WS-CDL* [11], or a technical infrastructure level, for example *SDL* [11, 20]. The approach seems to be useful for analysis of a network from the designer point of view. The description languages are supported by the simulation tools, for example modified *SSF Net* simulator [14, 15]. Still it is difficult to find the computer tools which are combination of model languages and Monte Carlo simulators [12, 16, 17].

4. Critical Situations

The working point of a unified network system is defined by specific values of functional parameters (resulting from the existing infrastructure – load capacity of commodity carriers and the available number of carriers, passing transfer limits, connection quality, availability and quality of handling equipment, route selection, etc.) and reliability (mean time to elements failures, the number of repair crews, the frequency and duration of traffic jams and other problems, machine renewal time, etc.). In practice, only some elements of the system's model may be treated as decision variables. For example, a system's designer may adjust carrier capacities to the actual needs of the task but very often, he or she has no possibility to choose the elements base on their reliability features. For example, it is possible to choose a better throughput of the connection, but it is no chance to change the parameters of this part of the network. The appropriate operating point of the network system may be achieved thankfully to the dispatching mechanisms and the actions of organizational nature as: choosing the number of carriers and/or the number of repair crews, bypassing a blocked (overload) by traffic connections, rescheduling, etc. Dispatching decisions concerning allocation of services (functionalities) and resources can define the system's reconfiguration necessary to accomplish the planned tasks.

The dependability analysis of network systems is carried out to assess the degree of risk associated with the implementation of task agreements. Note that in this case, the risk is defined and assessed as likely to ensure the system's performance under certain conditions. Another important issue is the evaluation of the impact of various system parameters on defined measures of performance (performability, dependability). Dependability synthesis of network systems is based primarily on proper selection of services and resources to fulfil the functional requirements defined by users' tasks (the so-called. input tasks) – see functional – reliable models [14, 15, 20].

Optimisation of system's synthesis is carried out based on the minimization of potential losses resulting from breach of contract. Since the parameters and decision variables of the process of network system's synthesis are determined by nominal values contained in the intervals of tolerance, though unlikely, is a scenario corresponding an operation point defined by the worst of circumstances (for example, the simultaneous maximum demand of tasks, the maximum number of long-term traffic jams, outbreaks caused by different matters, etc.). The decision variables and the parameters are very often treated as random variables within appropriate tolerance ranges. The operation point of the system may be defined together with a multidimensional solid of tolerance that is created at the appropriate confidence level.

The tolerance solid of the network system may be used as a basis for estimating the risk of system's faults. It is worth noting the difference between the intended ("built-in") redundancy (functional, reliable) and pseudo-redundancies as a result of random variables distributions, and therefore both the system's constructor and the dispatching mechanisms should exercise adequate caution in these situations. The set of system operation points forms a system's efficient operation area defined in n-dimensional hyperspace of system parameters and decision variables. The task of synthesis of the network system can be formulated as to ensure the global task performance for a specified number of carriers, choosing the appropriate delivery route and the costs do not exceed a fixed value. Figure 3 illustrates the problem of selecting the operation point of the network system taking into account the number of carriers and repair utensils. The actual system's quality is measured by the availability parameter.

The boundaries of the efficient operation area shall be determined on the basis of the acceptable costs of tasks, the maximum allowable repair time, and cost of used infrastructure. The boundaries can be set for the expected values – the hyper-planes of maximum costs of working system and the hyper-plane of the minimum, but still acceptable, system's availability. It is easy to notice that the efficient operation area may consist of many operating points, which are associated with different operating costs or risk of incorrect operation of the system. It is introduced a concept of a critical operation point of the system, i.e., such an operation point within the efficient operation area that the occurrence of a single hostile incident (e.g. damage of single system's element) causes a transient exit (e.g. for renewal time) beyond the area of efficiency and an additional hostile event that appears during the renewal time (e.g. a traffic jam on one of the used routes) leads to system's crush (e.g., interruption of the supply chain in a just at time operating system).

A subset of the critical operation points constitutes the so-called critical efficient operation area of the system (Figure 3) corresponds to critical system operation states. The critical system's state can be a simple consequence of change of "process parameters", such as raising the intensity of damage of the system's elements as a result of their use or the result of unfavourable combination of circumstances (adverse realization of random variables). For example, without necessarily changing the intensity parameter, too many carriers would be damaged at the same time, and repair crews would be overwhelmed. In extreme cases, it may lead to an avalanche of hostile events, or even to crash the system.

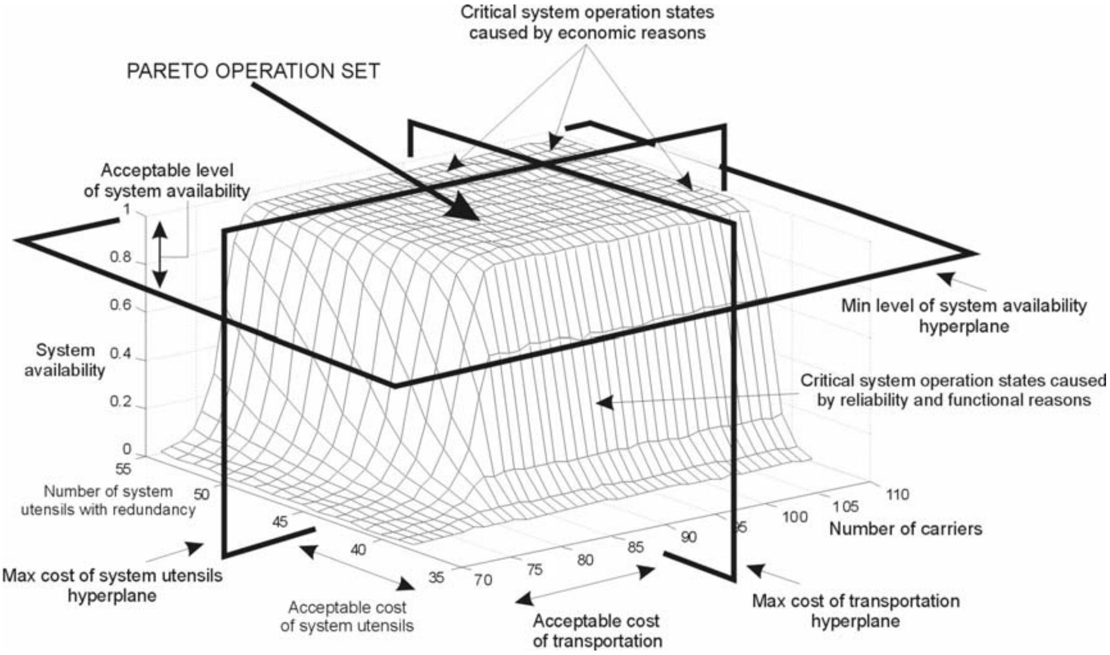
Critical system operation states caused by economic reasons
PARETO OPERATION SET
Acceptable level of system availability
1
0.8
0.6
System availability
0.4
0.2
0
55
Number of system utensils with redundancy
50
45
40
35
70
75
80
85
90
95
100
105
110
Number of carriers
Min level of system availability hyperplane
Critical system operation states caused by reliability and functional reasons
Max cost of system utensils hyperplane
Acceptable cost of system utensils
Acceptable cost of transportation
Max cost of transportation hyperplane

Figure 3. General idea of critical sets for network system

5. Conclusions

We have presented a formal model of sophisticated network system including reliability, functional parameters as well as the human factor component at the necessary level of detail. The model is based on the essential elements and features extracted from the ***Discrete Transport System (DTS)*** and the ***Computer Information System (CIS)***. We pointed the crucial conditions of the normal work of the system. The critical situation is described and discussed to create the Pareto set – guarantying the possible safety operating points for actual network system.

The proposed approach allows performing reliability and functional analysis of the different types of network systems, for example:

- determine what will cause a ”local” change in the system,
- make experiments in case of increasing volume of the commodity incoming to system,
- identify weak point of the system by comparing few its configuration,
- better understand how the system behaves.

Based on the results of simulation 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. Also the system could be analysed in case of some critical situation (like, for example, a few day tie-up [17]).

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.

Acknowledgment

The work reported in this paper was sponsored by a grant No. N N516 475940, (years: 2011-2014) from the Polish National Science Centre.

References

1. Al-Kuwaiti, M., Kyriakopoulos, N. & Hussein, S. (2009). A Comparative Analysis of Network Dependability Fault-tolerance, Reliability, Security, and Survivability. *IEEE Communications Surveys & Tutorials*, 11(2), 106-124.
2. Arvidsson, J. (2006). Taxonomy of the Computer Security Incident Related Terminology. *Telia CERT,* Retrieved May, 15, 2011, from (http://www.terena.nl/tech/projects/cert/i-taxonomy/archive/.txt)
3. Avizienis, A., Laprie, J.C., Randell, B. (April 2001). *Fundamental Concepts of Dependability*. Toulouse, France: LAAS-CNRS. (Research Report No. 1145, LAAS-CNRS).
4. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C. (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable and Secure Computing (TDSC)*, 1(1), 11-33.
5. Kołowrocki, K. (2004). *Reliability of Large Systems*. Amsterdam-Boston-Heidelberg-London-New York-Oxford-Paris-San Diego-San Francisco-Singapore–Sydney-Tokyo: Elsevier.
6. Kyriakopoulos, N., Wilikens M. (2001). *Dependability and Complexity: Exploring Ideas for Studying Open Systems, EN.* Brussels, Belgium: EC Joint Research Centre.
7. Lapie J. C. (Ed.). (1992). *Dependability: Basic Concepts and Terminology.* New-York, NY, Wien: Springer-Verlag,
8. Mazurkiewicz, J., Walkowiak, T., Nowak K. (2012). Fuzzy Availability Analysis of Web Systems by Monte-Carlo Simulation. In *Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence*, (pp. 616-624). Berlin, Heidelberg: Springer-Verlag.
9. Melhart, B., White, S. (2000). Issues in Defining, Analyzing, Refining, and Specifying System Dependability Requirements. In Proc. of the 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2000), Apr. 3-7, 2000 (pp. 334-340). Edinburgh, Scotland, UK: IEEE Computer Society.
10. Michalska, K., Walkowiak, T. (2008). Hierarchical Approach to Dependability Analysis of Information Systems by Modeling and Simulation. In Andre Cotton et al. (Eds.), Proceedings of the 2nd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2008), Cap Esterel, France, 25-31 August, 2008 (pp. 356 -361). Los Alamitos: IEEE Computer Society Press.
11. Michalska, K., Walkowiak, T. (2008). Modelling and Simulation for Dependability Analysis of Information Systems. In Jerzy Świątek et al. (Eds.), *Information Systems Architecture and Technology. Model Based Decisions* (pp. 115-125). Wroclaw: University of Technology.
12. Nowak, K. (2011). Modelling of Computer Systems – an Approach for Functional and Dependability Analysis. K. Kołowrocki, J. Soszyńska-Budny (Eds.), *Journal of Polish Safety and Reliability Association, Summer Safety and Reliability Seminars (SSARS 2011)*, 1, 153-161.
13. Walkowiak, T., Mazurkiewicz, J. (2008). Availability of Discrete Transportation System Simulated by SSF Tool. In Proceedings of International Conference on Dependability of Computer Systems, Szklarska Poreba, Poland, June, 2008 (pp. 430-437). Los Alamitos: IEEE Computer Society Press.
14. Walkowiak, T., Mazurkiewicz, J. (2008). Functional Availability Analysis of Discrete Transportation System Realized by SSF Simulator. In Proceedings of the 8th International Conference

'Computational Science – ICCS 2008', part I, Krakow, Poland, June 2008, (pp. 671-678). Berlin, Heidelberg: Springer-Verlag.

15. Walkowiak, T., Mazurkiewicz, J. (2010). Algorithmic Approach to Vehicle Dispatching in Discrete Transportation Systems. In Jarosław Sugier et al. (Eds.), *Technical Approach to Dependability* (pp. 173-188). Wroclaw: Wroclaw University of Technology.
16. Walkowiak, T., Mazurkiewicz, J. (2010). Functional Availability Analysis of Discrete Transportation System Simulated by SSF Tool. *International Journal of Critical Computer-Based Systems*, 1(1-3), 255-266.
17. Walkowiak, T., Mazurkiewicz, J. (2010). Soft Computing Approach to Discrete Transportation System Management. In *Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence*. vol. 6114 (pp. 675-682). Berlin, Heidelberg: Springer-Verlag,
18. Volfson, I.E. (2000). Reliability Criteria and the Synthesis of Communication Networks with its Accounting. *J. Computer and Systems Sciences International*, 39 (6), 951-967.
19. Xiaofeng, T., Changjun, J., Yaojun, H. (2005). Applying SOA to Intelligent Transportation System. In Proceedings of the IEEE International Conference on Services Computing, Vol. 2, July, 11-15, 2005 (pp. 101-104). Orlando, Florida: IEEE Computer Society.
20. Zamojski, W., Caban, D. (2005). 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), 2005, (pp. 2087-2090). Gdynia-Sopot-Gdansk: A. A. Balkema.
21. Zhou, M., Kurapati, V. (1999). *Modelling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach.* London, UK: World Scientific Publishing.
22. Zhu, J., Zhang, L.Z. (2006). A Sandwich Model for Business Integration in BOA (Business Oriented Architecture). In Proceedings of the IEEE Asia-Pacific Conference on Services Computing (APCSC), 2006 (pp. 305-310). Washington, DC: IEEE Computer Society.