Zamojski Wojciech

Mazurkiewicz Jacek
Wrocław University of Technology, Poland

From reliability to system dependability – theory and models

Keywords

system reliability, system dependability, networks, services, functional and dependability models

Abstract

The paper presents a novel approach to system dependability problem. The analysed systems and networks - characterised by different sets of features - are considered as a union of all their resources essential for the predicted tasks realisation. The system dependability is discussed with respect to the occurrence of incidents and treats that may cause to damage the system resources and - in consequence - to the collapse of the executed tasks. The maintenance policy system is based on two main concepts: detection of unfriendly events and system responses to them. The proposed analysis is realised from the user's point of view, focusing on functional features described by business services available in the system.

1. Introduction

The contemporary systems are created as very sophisticated products of human idea characterized by the complex structure. On the other hand the systems combine two types of resources: technical (engineering stuff) and information (algorithms, processes and management procedures). The systems are human-controlled and computer-aided devices. The reliability parameters of the system resources are very screwed-up – so the analysis (we can also say: synthesis) of contemporary systems needs adequate models and calculation methods [20], [21].

During more than 60 years the reliability theory was altered from the reliability of single and separated objects (elements) considered only two states ("efficient work", failure) to the contemporary dependability of systems or even the dependability of service nets. The indicated development of the reliability theory is the consequence of expanding the event sets taken in the consideration for the reliability models. The present system dependability theory considers not only classical reliable events (failures or repairs) but tries to combine all types of the faults generated by the system resources (hardware, algorithms, human-factor) and the environmental features which may disturb the operable state (attacks – for example). The main goal of the reliability analysis and the measures calculation is to convert the discussion focused on the reliability function of elements (or structures created by the element sets) into the task performance or efficiency estimation. The tasks are realised according to the system services [2], [3].

The paper presents our (W. Zamojski and his scientific staff) point of view on the system reliability description. We call the approach as the functional-reliability models. The computer systems analysis is the root for our elaboration but we believe it is useful for modelling of the wider spectrum of systems which realise tasks based on fully or partially available resources. We think about a discrete transport system or power management systems for example. Models of the real contemporary systems are complicated – a lot of the different events described by the different distribution functions. We propose to use non-classic solution for the necessary reliability measures evaluation based on the Monte-Carlo simulation technique [19], [20], [21], [22], [23]. This way we are able to relax very strict assumptions about the kind of distribution previously fixed by the Markov model approach. The functional-reliability models should be – before the start of the simulation process – automatically translated into a computer program. The special

languages and computer tools for such task are known and available [12], [30], [31], [32].
The computer and software equipment allows making the reliability theory more sophisticated. The simulation technique is the real chance to operate with large systems – where the number of elements is significant. The elements can be described by different sets of features. We can observe – in parallel – large number of events in quite long time-periods. This way we can collect data sets to very detailed presentation of the system life. Based on the data we are able to elaborate the formal theoretical approach of the system dependability [26], [27], [29]
In the section 2 the essential properties of the contemporary systems are presented. The reliability and dependability models of the system and their usefulness for the analysis are discussed in the next sections.

2. Systems

2.1. A system and its tasks

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, that may even be modified by the studied systems.
It can be distinguished three main elements of any system: users, services (functionalities) and technological resources. 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 resources [1], [2], [3].
The system S_C is a configuration of technological resources (hardware) *H*, information resources (software) *SP*, human-factor *M*, management system (operating system) *MS*, tasks (functions) *J* and system events E_S:

$$S_C = \langle H, SP, J, M, MS, E_S \rangle \qquad (1)$$

A technological resource 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 understood little bit different: she or he can be finding as: a system operator, a service person, a system manager (administrator) etc. [33], [34]
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 reactions to them (technological and information renewals).
The interactions of these components are expressed as task configurations $S_M(j)$. Each active task in the system has its own configuration, i.e.

$$S_C(j) = \langle H(j), S_S(j), S_M(j) \rangle; \quad j \in J, \qquad (2)$$

where:
$H(j)$ – the subset of technological components used in processing the task *j*,
$S_S(j)$ – the subset of information elements,
$S_M(j)$– the subset of human users/operators.

Task configurations changes when the tasks are being processed. The changes are determined by the software management, reacting with the system users. Some changes may be the result of detecting system faults and reacting to them. This is called system reconfiguration [24], [30]
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 [12], [13]

2.2. Incidents

The system incidents may be classified as unintentional damages generated by faults of the hardware, software or human-factor and intentional events aimed at harming the information resources and system processes. Very often incident is a result of a broadcast attack that is not addressed to a fixed entity (device, machine, truck, lathe, computer, and network) but to all anonymous entities (systems, computers or networks). This kind of attack is called virus. An incident may be "insignificant" if its consequences are easily removed from the system. Sometimes an incident may have a more serious impact on the system behaviour: it may escalate to a security incident, a crisis or a catastrophe.

If a fault appears during the task execution then the system – based on the decision of its managing system and/or its operators – starts the renewal processes. Time of the technological renewal activities and the time of informational renewals are added to the nominal time of the task so a real time of the task duration is longer (*Figure 1*). The real duration time of the executed tasks depends on kind of faults; hardware failures need both renewals; technological and information but removing of consequences of human errors or software ones is very often limited only to the information renewal process [2], [3], [9], [26].

2.3. Maintenance policy

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 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 manager) are build in every real system [2], [3], [24], [26].

The maintenance policy is based on two main concepts: detection of unfriendly events and system 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 responses incorporate the following procedures:

- detection of incidents and identification of them,
- isolation of damaged resources (hardware and software) 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 in the *Figure1*.

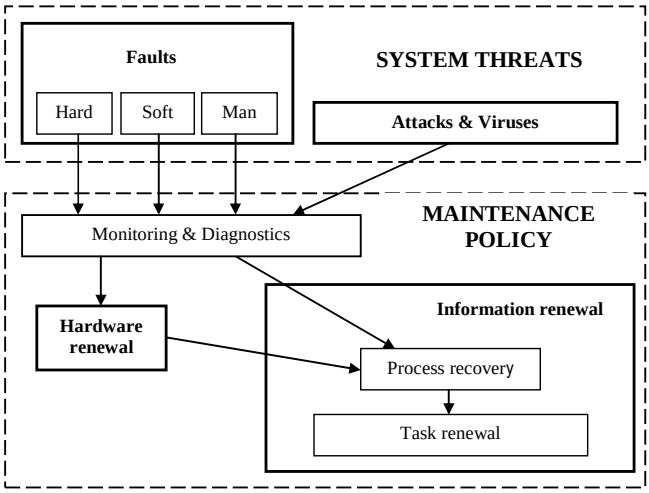

Figure 1. Threats and system reactions [32]

3. Reliability models and evaluation methods

When we are looking at reliability history we may point out a few development reliability stages (classes) spread from the classical reliability till the contemporary dependability. Reliability development stages are defined based on the considered faults and the renewal processes which are leaded to the adequate mathematical models and the calculation methods.

The basic principle of reliability stage is the generalisation of the definition of system (object) operability: the system is regarded as operational if it is capable to realise the required tasks in the required time period under given environmental conditions.

The system tasks are defined at the different levels from being in an efficient state till a function (jobs, works) sequence realised on chosen base of the system resources with assumed performance and time parameters.

As follows from this generalised definition, some failures of the equipment may cause degradation of system efficiency such that it is still operational, being capable of fulfilling its tasks in the required time.

3.1. Probability models – unrenewal systems – "to be or not to be"

The classical reliability theory considers non-repaired objects (elements) that have only two states: efficient work and broken. The first state means that an element realises its duties (is "alive") and in the second case a failure has occurred and the element is broken (is "dead"). The system is considered as a set of elements which creates structures (serial, parallel, mixed serial-parallel, *k* out of *n*, etc.). The structures work only in two states: efficient and broken again.
This kind of mathematical reliability models (called classical reliability models) is very popular among engineers and mathematicians. The classical models are basis for evaluation such reliability measures as probability of efficient work (*R(t)* – reliability function) or mean time to first failure (*MTTFF*) for different failure rate functions. The functions often are difficult to compute explicitly, so upper and lower bounds are estimated [9], [10], [24]
Unfortunately, usefulness of the classical models is limited in practice of the contemporary systems. The real systems are frequently build of renewal elements which can operate in the environment that "produce" malfunctions or attacks and very often system reliability analyses are not only limited to reliability parameters.

3.2. Markov models – renewal systems

The next step in reliability development is connected with the mathematical renewal theory.
Because majority of the real elements and systems is repaired so we are especially interested in two reliability problems: if the system is able to efficient work in fixed future moment or period and how long time-period is lost for the system renewal. The efficient working time depends on the distribution functions of the time to failure for each element and the whole system; "a system reliability structure" has also the significant influence. The system renewal time depends on the organisation of maintenance described by such parameters as distribution functions of renewal time, a number of servicepersons and its relation with a number of resources that may be broken and must be renewal (maintenance person problem), redundancy, etc.
The renewal systems are modelled as a *State - Transitions* directed graphs (it is called *ST-graph* or *ST-model*). The system events (failures, renewals) are represented by transitions (arcs) connected to the states defined based on the system reliable properties. It is assumed that *ST-graph* transitions model allows following the single event at the moment. It means that at single moment only one element is broken or renewal.
The *ST-graphs* are very usefully tools for graphical modelling and analysing life of the renewal systems, especially for small systems (with a small number of states). When the number of states increases then the construction of the graph is complicated.
States of the *ST-graph* are dividing into three sets: a set of efficient working states (all system functions are correctly executed), full unreliable states (no functions are correct realised) and states of partial efficient working.
When time distribution functions of failures and repairs are exponential (or are described by a sum of the exponential distribution functions) then the *ST-graph* is considered as the Markov process (or the Semi-Markov process) and then it is easy (Chapman - Kolmogorov equations) to calculate probability that the system will be in each separate states and next to find needed reliability measures. Of course assumptions about exponential distribution functions are very hard and to far of real life.
The *ST-model* is the most popular and useful methodology used for the modelling of systems. The *ST-models* are used in reliability measure evaluation also based on the Monte Carlo simulations.
If we try to ascribe the classical Markov model to the real life observations we can point the following matters:

1. The elements are characterised by more than two states: dead or alive. Their live can be find as more "exciting" – so the Markov model offers insufficient and trivial state description.
2. The reliability structure of the system is very sophisticated and complicated. The Markov model does not give the real chance to describe it.
3. The elements and the whole system can be discussed as more or less efficient – so the "reliable situation" is not bi-level. We can introduce the function of efficiency. Barlow-Prosnan presents the example of military radar system [20].

In general – we can conclude:

1. The Markov processes use very strict assumptions (only exponential distributions or approximations by a sum of expo distributions) – it is hard to reconcile them in practice.
2. The engineering point of view cannot agree with the Markov modelling approach.

3.3. Performability models – functional and reliability properties of systems

The basic principle of this class of models is the generalisation of the definition of system operability: the system is found as operational if it is capable to realise the required tasks in the required time period satisfying given environmental conditions. It means some failures of the equipment may cause degradation of system efficiency, but it is still operational and able to fulfil its tasks in the required time-period.
A functional – reliability model for a system engineering problems is 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 are divided into two main classes: functional events and reliable (including maintenance) events. The practice finds the classification very often difficult to introduce because of a system reaction for an event involving 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]. This way considered model of systems is called *performability* model or *functional-reliability* model [16], [26].
If the functional–reliability model is built as the *ST-model* – the set of the system states is determined by the states of all resources involved in tasks realised at the moment. The system resource allocations are dynamic, modified due to the incoming tasks, occurring incidents and system reactions (especially reconfiguration). If we think about the discrete transport systems we can decide for the number of vehicles operated for single task, we can reconfigure the routes or we can combine the vehicles to operate for more than single task. It is possible because the commodities are addressed [27].

3.4. Dependability models - systems in active environments

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) [2], [3].
The concurrent systems are very often considered as a large network, for example: information, transport or electricity distribution systems.
The system 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 state alterations) and *maintainability* (ability to undergo repairs and modifications) [1], [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 requirements. We can easy find some quantitative and qualitative parameters of user tasks [11], [26].
The system 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 technological infrastructure or the part o 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 build 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 technological infrastructure that means reliable properties of the system 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 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 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 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 dependability (and safety) and on the economic conditions (first of all, the cost of downtime and lost processing/computations) [11].

3.5. Simulation models

The complexity of the concurrent systems necessitates the construction of new classes of the mathematical models. The progress in the computer engineering has made it feasible to avoid many of the traditional simplifying assumptions, e.g. it is no longer necessary to make the basic Markov assumptions that events are single, independent and with no memory. The computer simulation makes it possible to consider system events with any distributions and with various dependencies among the streams of events (stochastic processes) [21], [22].

Taking this into consideration, it is necessary to develop methods and meta-languages to model the systems, characterised by the multiple concurrent processes with interdependencies. Perhaps this may be achieved by combining the *ST-models* with Petri nets and descriptive languages (e.g. UML) [12], [33]. Set theory models might be used to define the components of the modelled systems.

It is also necessary to produce software capable of analysing the simulation models, automatically choosing the methods and tools needed to perform full simulation testing.

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 find 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 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], [6, [25] or Business Oriented Architecture [25], 34], and a lot of languages for network description on a system choreography level, for example *WS-CDL*, or a technical infrastructure level, for example *SDL* [7], [12]. 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 [21], [22]. Still it is difficult to find the computer tools which are combination of model languages and Monte Carlo simulators [19], [22], [23].

4. Dependability models as a services net models

4.1. Proposed model discussion

A *services network* is a system of business services that are necessary for user (clients) tasks realisation process. The services net are built based on the technical infrastructure (*technological resources*) and the *technological services* which are involved into a task realisation process according to decisions of a management system. The task realisation 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 realisation steps is called as *choreography*.

The 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 realised by the network [29].

We can find more general definition of the system (1) introducing the idea of the net of services. It is described at the upper level of abstraction: the single service corresponds to the sequence of tasks and jobs realised using the net resources.

The service net could be defined as a quintuple:

$$SNet = \langle J, BS, TR, MS, C \rangle \tag{3}$$

where:

J – a set of tasks generated by users and realised by the service network,

BS – a set of services which are available in the considered network,

TR – technological infrastructure of the network which consists of technical resources as machines/servers, communication links etc,

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

C – a network chronicle, defined by a set of all essential moments in a "life" of the network.

The task $J^{(i)}$ is understood as a sequence of actions and jobs performed by services network in a purpose to obtain desirable results in accordance with initially predefined time schedule and data results.

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.
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 realised based on a few separated sets of functionalities with different costs which are the consequences of using different network resources.
The management system of services network allocates the services and network resources to realised tasks, checks the efficient states of the services network, performs suitable actions to locate faults, attacks or viruses and minimises 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 persons 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 task realisation 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: the services management and the resource management. 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 accidences [2], [3], [29]
At the beginning of a task realisation procedure the task $J_{IN}^{(i)}$ is mapped into the network services and a subset of services $BS_S^{(i)}$ necessary for the task realisation according to its postulated parameters is created: $J_{IN}^{(i)} \rightarrow BS_S^{(i)}$, s = 1, 2, ... Next, a demand of technical resources for each service realisation is fixed: $BS_S^{(i)} \rightarrow R_n^{(i,s)}$, n = 1, 2, ... In a real services network the same task is often realised using the 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 are checking for each service.
This way a few task configurations service configurations, additionally described by appropriately defined cost parameters, may be fund for the *i-th* task realisation.

4.2. Safety aspects discussion

The safety problems related to the computer driven systems combine the set of means to estimate and to control the risk of systems, nets, data channels usage in case of integrity, availability and confidentiality of the work.
There are a lot of international standards to describe the safety of the computer driven systems. First of all we can find the Common Criteria methodology [4] included into *ISO/IEC 15408* or the huge programs Cybersecurity for Critical Infrastructure Protection in the USA [17], [18]. The European Union also promotes scientific elaborations close to the dependability problems: *POSITIF* Policy-based Security Tools and Framework (*IST-2002-002314*), *IRRIIS* Integrated Risk Reduction of Information-based Infrastructure Systems (*IST-2004-027568*), *DESEREC* Dependability an Security by Enhanced Reconfigurability (*IST-2004-026600*), ITEA-ENERGY Empowered Network Management (*ITEA 04024*) [5, 28].
The dependability in the contemporary world is very close to the *SOA* – Service Oriented Architectures, the *SOM* – Service Oriented Management, and the *SCA* – Service Component Architecture, approaches. The problems of the service choosing and the service management are the main goal of these technologies. One of the available *SOA* definitions says that it is a solution for a sophisticated business process spreads at non-

trivial space of heterogenic systems having different owners. The *SOA* approach is realised using Web Services: *SOAP*, *WSDL*, *UDDI* standards [6], [8], [25]

The functional and dependability models are created as operations in the sets of: tasks (ready to be realised and already realised), devices, software, management (including human operators and owners), time described by a chronicle. The functional configuration of the system is the sets of means necessary to the task realisation. The resources are allocated dynamically and are driven by the set of new tasks and the stream of failures. The system reaction – the proper maintenance approach according to the actual functional and reliability situation generates very sophisticated models. It is rather easy to describe them using high-level languages as *UML* for example. But it is still unsolved problem how to translate the verbal described model into set of business services and know, ready-to-use languages dedicated to system and hardware description as *SDL* and *WCSDL*. The problem of the automatic – made by computer – translation of the functional and dependability model into service and hardware level is necessary if we try to verify and analyse real systems which operate with business services [14], [15]. The problem is more sophisticated if we introduce the means to improve the safety level and the required maintenance politics. The problem seems to be crucial for computer networks, banking, and general e-business solutions [29].

5. Conclusion

The paper discusses different aspects of the functional – dependability model of the system and service networks. The formal model consists of a quintuple of proper sets of elements. Of course there are a lot problems described by a big number of services and technical resources that are mapped to the many concurrent realised tasks. A lot of possible maintenance politics, which can be find as network reactions on hypothetic or real faults and threats, complicate the proposed models, especially if costs of maintenance rules are considered. Additional problems are consequences of possibilities to realise the task based on the different services and resources, which means – various costs. In general we can say that dependability models analysis drive our approach to the safety matters of the system.

Acknowledgment

Work reported in this paper was sponsored by a grant No. N N509 496238, (years: 2010-2013) from the Polish Ministry of Science and Higher Education.

References

[1] Arvidsson, J. (2006). *Taxonomy of the Computer Security Incident Related Terminology*. Telia CERT (http://www.terena.nl/tech/projects/cert/i-taxonomy/archive/.txt)

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

[3] Avižienis, A., Laprie, J., Randell, B. & Landwehr, C. (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computin*, Vol.1, 11–33.

[4] Common Criteria. ISO/IEC15408

[5] GAO-04-321 (2004). TECHNOLOGY ASSESSMENT. Cybersecurity for Critical Infrastructure Protection. US GAO, Washington.

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

[7] IFIP WG10.4 on Dependable Computing and Fault Tolerance, http://www.dependability.org/.

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

[9] Kołowrocki, K. (2004). *Reliability of Large Systems*. Elsevier, Amsterdam - Boston - Heidelberg - London - New York - Oxford - Paris - San Diego - San Francisco - Singapore - Sydney - Tokyo.

[10] Kopociński, B. (1977). Zarys teorii odnowy i niezawodności. PWN Warszawa.

[11] Mortier, R., Isaacs, R. & Barham, P. (2006). Anemone: Using End-Systems as a Rich Management Platform. Microsoft Technical Report, MSR-TR-2005-62.

[12] Michalska, K. & Walkowiak, T. (2008). Hierarchical Approach to Dependability Analysis of Information Systems by Modelling and Simulation. *Proc. of the 2008 Second international Conference on Emerging Security information, Systems and Technologies. SECURWARE*. IEEE Computer Society, Washington, DC, 356-361.

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

[14] Oppenheimer, D. et al. (2002). ROC-1: Hardware Support for Recovery-Oriented Computing. *IEEE Trans. On Computers*, Vol. 5 no 2.

[15] Patterson, D. et al. (2002). Recovery Oriented Computing (ROC): Motivation, Definition,

Techniques, and Case Studies. Berkeley: Comp. Sc. Technical Report.

[16] Ross Sheldon, M. (1985). *Introduction to Probability Models* (Third Edition). Academic Press, Inc. Orlando Florida.

[17] Report to the President (2005). Cyber Security: A Crisis of Prioritization. Executive Office of the President of the US.

[18] Supplement to the President's Budget for Fiscal Year 2007 (2006). The Networking and Information Technology. *Research and Development Program*. Executive Office of the President of the US.

[19] Walkowiak, T. & Mazurkiewicz, J. (2009). Analysis of Critical Situations in Discrete Transport Systems. *Proc. of International Conference on Dependability of Computer Systems*, Brunow, Poland, June 30-July 2, 2009. Los Alamitos: IEEE Computer Society Press, 364–371.

[20] Walkowiak, T. & Mazurkiewicz, J. (2008). Availability of Discrete Transport System Simulated by SSF Tool. *Proc. of International Conference on Dependability of Computer Systems*, Szklarska Poreba, Poland, June, 2008. Los Alamitos: IEEE Computer Society Press, 430–437.

[21] Walkowiak, T. & Mazurkiewicz, J. (2008). Functional Availability Analysis of Discrete Transport System Realized by SSF Simulator. *Computational Science* – ICCS 2008, 8th International Conference, Krakow, Poland, June 2008. Springer-Verlag, LNCS 5101, 2008. Part I, 671–678.

[22] Walkowiak, T. & Mazurkiewicz, J. (2010). Functional Availability Analysis of Discrete Transport System Simulated by SSF Tool. *International Journal of Critical Computer-Based Systems*, Vol. 1, No 1-3, 255–266.

[23] Walkowiak, T. & Mazurkiewicz, J. (2010). Soft Computing Approach to Discrete Transport System Management. *Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2010. Vol. 6114, 675–682.

[24] Volfson, I.E. (2000). Reliability Criteria and the Synthesis of Communication Networks with its Accounting. *Journal Computer and Systems Sciences International*, Vol. 39 (6), str. 951-967, Nov. – Dec.

[25] Xiaofeng, T., Changjun, J. & Yaojun, H. (2005). Applying SOA to Intelligent Transportation System. *IEEE International Conference on Services Computing*, 101–104 vol. 2.

[26] Zamojski, W. (2005): Model funkcjonalno-niezawodnościowy systemu komputer-człowiek. (*Functional-Reliability Model of a Computer-Man System)*. Praca zbiorowa pod redakcją Wojciecha Zamojskiego Inżynieria komputerowa. WKiŁ, Warszawa.

[27] Zamojski, W. (2007). *Systemy transportu dyskretnego. Niezawodność. Modele.* Praca zbiorowa pod red. W. Zamojski. WKŁ Warszawa.

[28] Zamojski, W. & Caban, D. (2003). Trends in the Theory and Engineering of Reliability Applied to the NBIC Technology. *Proc. of The 3-rd Safety and Reliability International Conference to Safer Life and Environment KONBIN 2003*, Gdynia.

[29] Zamojski, W., & Walkowiak, T. (2010). Services net modelling for dependability analysis. *In-Tech.*, 1-17.

[30] Zamojski, W. & Caban, D. (2005). Assessment of the Impact of Software Failures on the Reliability of a Man-Computer System. *Proc. Of European Safety and Rel. Conference ESREL 2005*, A.A. Balkema Publ.

[31] Zamojski, W. & Caban, D. (2006). Introduction to the Dependability Modelling of Computer Systems. Dependability of Computer Systems DepCoS - RELCOMEX '06,100–109.

[32] Zamojski, W. & Caban, D. (2007). Maintenance Policy of a Network with Traffic Reconfiguration. Dependability of Computer Systems DepCoS - RELCOMEX '07, 213–220.

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

[34] Zhu, J. & Zhang, L.Z. (2006). A Sandwich Model for Business Integration in BOA (Business Oriented Architecture). *Proc. of the 2006 IEEE Asia-Pacific Conference on Services Computing.* APSCC. IEEE Computer Society, Washington, DC, 305–310.