Dependability of services networks

Keywords
dependability, networks, services, functional-dependability models, Petri nets

Abstract
In the tutorial paper systems and networks are considered as a union of all their resources essential for the realization of predicted tasks. System dependability is discussed with respect to the occurrence of incidents and treats that may cause damage to the system resources and, in consequence, to the executed tasks. The maintenance policy system is based on two main concepts: detection of unfriendly events and system responses to them. It is proposed to analyse the network system from the functional and user point of view, focusing on business service realized by a network system. Services networks are modelled as the Petri net models which may be useful for analyses of dependability parameters and for setting up a maintenance policy of the services net.

1. Introduction
Network technologies are being developed for many years. Most of large technical systems could be seen as a kind of network, for example: information, transport or electricity distribution systems. Networks are modelled as directed graphs with nodes, in which commodities and information media are being processed, and arcs as communication links (telecommunication channels, roads, pipelines, conveyors, etc.) for media transportation. Resources of networks could be divided into two classes: services (functionality resources) and technical infrastructures (hardware and software resources). Dependability of the system is its ability to execute correctly system functions (tasks, jobs) in anticipated time, in assumed work conditions (environment), and in presence of threats, hardware failures or software and human faults.

The system dependability is 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].

It is proposed to analyse the network system from the functional and user point of view, focusing on business service realized by a network system [11]. 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). 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.

It is assumed 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.

The dependability of a system is the ability to deliver service that can be justifiably trusted [2]. Sometimes the system dependability is considered as its survivability, that is its ability to execute correctly all system functions in spite of possible threats (outside
or inside attacks) and system incidences (hardware failures, software and human faults). Since it is impossible to prevent all incidents and attacks on a system, it is essential to react quickly (and consistently) in order to stop the proliferation of their consequences within the system. In case of serious incidents or attacks, the system must still provide some essential services. Thus, survivability is defined as “the capability of a system to fulfil its mission, in a timely manner, in the presence of attacks, failures, or accidents”.

2. Incidents and network reactions

An incident is an unintended system event that might lead to some disruptions in the system behavior. The incident may cause some damage to the system resources; hardware, software or information and, in consequence, it may disrupt the executed processes. If a fault appears during the task execution then the system on the base of decision of its operating system or its operator (man) starts renewal processes. Time of technological renewal activities and time of informational renewals are added to the nominal time of the task so a real time of the task duration will be longer (see 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 only limited to the information renewal process.

The system incidents may be classified as unintentional damages generated by faults of the hardware, software or men 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 (computer, network) but to all anonymous entities (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 behavior: it may escalate to a security incident, a crisis or a catastrophe.

3. Maintenance policies

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 behavior 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 between incidents and reactions of the system are shown on the Figure 1; incidents are limited to attacks on system processes and information resources (the possibility of hardware damages resulting from these intrusions is not accounted for).

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 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 should be formulated on the basis of the required levels of system dependability (and safety) and on the economical conditions (first of all, the cost of downtime and lost processing / computations) [11].

Maintenance rule \(mr_j, j=1,2,...\) is a chain of decisions about allocation of system 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 node processor or communication interface. These local operations may have impact on the whole system, e.g. if a communication channel is down for a few minutes, then rates of data traffic of the system may violently change [9].
Cost of maintenance rule. Cost of the j-th maintenance rule ($c_{mr,j}$) is defined as all costs of ventures used to ensure required level of operation of a renewed part of the system. These costs may be evaluated in many ways: as cost of replacing broken computer for a new one, as servicemen costs, as time lost during renewal, etc [12], [13]. For example, a cost of the maintenance rule may be evaluated as

$$c_{mr,j} = Ac_{mr,j} + Oc_{mr,j}$$

where

$Ac_{mr,j}$ - cost of additional system resources (hardware, software, servicemen, etc),

$Oc_{mr,j}$ - operational cost of implementing the maintenance rule.

The cost of additional system resources is evaluated as all costs that are incurred to make the maintenance rule possible (hardware and software investments, staff hire). They are incurred regardless whether the incident occurs or how often it occurs.

The operation cost of the maintenance rule may be considered as the cost of data lost during the network renewal and the unit costs of renewal procedures (cost of spare parts, cost of a service etc). The maintenance rules may be applied many times during time period $[0,T_e]$, so the real cost of the rule should be estimated as a mean value

$$c_{mr,j}(T_e) = E_{T_e} \left[ Ac_{mr,j}(t) + Oc_{mr,j}(t) \right]$$

where

$E_{T_e} \left[ \cdot \right]$ - the mean value computed over the exploitation time $[0,T_e]$ [13].

Local maintenance policy may be based on the analysis of the pairs: maintenance rule and cost of its execution

$$mp_j = \{ mr_j, c_{mr,j} \}.$$

Renewal of a system component may often be realized in more than one way, at different levels of cost. So, there is a set of pairs and the local maintenance policy is defined on this set. The choice does not necessarily depend on the comparison of costs, it may be determined by actual availability of spare parts or services, or on an arbitrary decision of an administrator.

System maintenance policy (MPS) is a triple of sets:

$$MPS = \{ SI, MP, D_{mp} \}$$

where

$SI$ - system incidents,

$MP$ - maintenance policies,

$D_{mp}$ - rules characterizing the impact of each maintenance rule on the overall system performance (system cost).

A very simple example of the system maintenance policy is demonstrated in Table 1, where all the foreseeable incidents are listed in the first column. For each incident, various feasible maintenance rules are listed in column 2, with corresponding local costs (column 3). The last column is the most important, since it gives grounds for adopting maintenance decisions based on the analyses of the whole system. For example, in a network of computers, a chosen local maintenance policy may have a huge and diverse influence on various parts of the network, so locally the cheapest local maintenance policy does not have to yield the best global solution. The global maintenance policy has to depend on the dependability measures of the system and the impact of local maintenance policies on them.

<table>
<thead>
<tr>
<th>Incidents</th>
<th>Maintenance policy rule</th>
<th>Maintenance policy cost</th>
<th>System cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k-1$</td>
<td>$mr_k(1)$</td>
<td>$c_{mr_k(1)}$</td>
<td>$d_{mr_k(1)}$</td>
</tr>
<tr>
<td>$k$</td>
<td>$mr_k(2)$</td>
<td>$c_{mr_k(2)}$</td>
<td>$d_{mr_k(2)}$</td>
</tr>
<tr>
<td>$k+1$</td>
<td>$mr_k(j)$</td>
<td>$c_{mr_k(j)}$</td>
<td>$d_{mr_k(j)}$</td>
</tr>
<tr>
<td>new incident</td>
<td>empty at this moment</td>
<td>empty</td>
<td></td>
</tr>
</tbody>
</table>

4. Tasks and services networks

It is proposed to concentrate the dependability analyse of the networks on fulfilling of user requirements. Therefore, it should take into consideration following aspects:
• specification of the user requirements described by task demands, for example certainty of results, confidentiality, desired time parameters etc.,
• functional and performance properties of the networks and their 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 [3], [6] or Business Oriented Architecture [9], [10], and a lot of language for network description on a system choreography level, for example WS-CDL, or a technical infrastructure level, for example SDL [7]. These propositions are useful for analysis of a network from the designer point of view and they may be supported by simulation tools, for example modified SSF.Net simulator [7], but it is difficult to find a computer tools which are combination of language models and Monte Carlo simulators.

It can be distinguished three main elements of any network system: users, services and technical resources. As it presented in the Figure 2 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.

![Figure 2. Tasks mapping into network resources](image)

Therefore, following problems should be taken into consideration:
• description and mapping a service net on existed net resources for each moment of its using;
• a prognoses process of the service net behavior in a real life conditions – definition and selection of measures;
• finding relations between measures/criteria and functional, performance and reliability parameters of the service net;
• evaluation methods of choose measures of the service net;
• decision process of maintenance organization - decision steps as a reaction on appeared events, specially on threats;
• 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 on Figure 3.

5. Functional – dependability models

The ST model (State - Transition model) is the most popular and useful methodology used in modeling 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.
The functional – reliability model [11] 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 $E_S$

$$S_C \subseteq H \times SP \times J \times M \times MS \times E_S$$

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 $j^{(i)}$;

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

and

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

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 [8]. Because these reasons considered model of services network will be called performability model or functional-dependability model [11].

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).

6. A services net model

A services network 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.

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

The service net could be defined as a tuple:

\[ SNet = \{ J, BS, TR, MS, C \} \]

where:

\[ J = \{ J^{(i)}; i = 1, 2, \ldots \} \] - a set of tasks generated by users and realized by the service network,

\[ BS = \{ BS^{(b)}; b = 1, 2, \ldots \} \] - a set of services which are available in the considered network,

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

\[ MS \] - management system (for example - operating system),

\[ C = \{ c_i; t = 1, 2, \ldots \} \] - a network chronicle, defined by a set of all essential moments in a “life” of the network.

6.1. 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 \( J^{(i)} = \{ J^{(i)}_N, J^{(i)}_O \} \) may be defined as an ordered pair of so called input task \( J^{(i)}_N \), which is described by the input parameters (postulated results and prognosis time schedule) and the corresponding output task \( J^{(i)}_O \) (real results and real time schedule).

The input task is defined as the triple:

\[ J^{(i)}_N = \{ R^{(i)}_P, A^{(i)}, C^{(i)}_P \} \]

where

\[ R^{(i)}_P \] - postulated results of the \( i \)-th task execution,

\[ A^{(i)} = A^{(i)}(R^{(i)}_P, C^{(i)}_P) \] - a sequence of actions and works necessary to obtain postulated results in planned time. The \( A^{(i)} \) may be described by a flowchart of actions and works, and its realization depends on an availability of network services and technical resources,

\[ C^{(i)}_P \] - postulated chronicle of the task realization.

The output task is defined as the pair

\[ J^{(i)}_O = \{ R^{(i)}_P, C^{(i)}_P \} \]

where:

\[ R^{(i)}_P \] - real results of the \( i \)-th task execution,

\[ C^{(i)}_P \] - real chronicle of the task realization.

The postulated results and chronicles are defined with assumed tolerance intervals ( \( R^{(i)}_P \leq R^{(i)}_P \leq R^{(i)}_P \)) and ( \( C^{(i)}_P \leq C^{(i)}_P \leq C^{(i)}_P \)) and when the real results and chronicles are inside the intervals ( \( R^{(i)}_P \subset [R^{(i)}_P, R^{(i)}_P] \) and \( C^{(i)}_P \subset [C^{(i)}_P, C^{(i)}_P] \)) then the task is assumed to be correctly realised.

6.2. 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.

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

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

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 $\{ F_k^{(b)}; k1 = 1,2,... \}$, $\{ F_k^{(b)}; k2 = 1,2,... \}$ ... with different costs which are the consequences of using different network resources.

### 6.3. 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.

### 6.4. Management system

The management system of services 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. In many situations a team of men and the management system have to cooperate in looking for adequate decisions (for instance under a heavy attack or when a new virus is disclosed).

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.

### 6.5. 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).

### 6.6. 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 accidents.

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

### 7. The Petri net model

Petri Nets 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: $PN = \{ P, T, A \}$ where
A state of the net, described by marking (tokens localization in the places) represents sufficient conditions for arising new events of a net’s life. Net’s events may be divided into many classes, for example functional, reliable or maintenance events, deterministic or probabilistic ones etc. The mention classification depends on assumed criteria.

The Petri net model of the $i^{th}$ task realization ($J(i)$) is shown on the Figure 3. It is assumed the input task ($J^{(i)}_{IN}$) 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 3) 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...

**Figure 4.** An example of the Petri net model of a services network
time $\tau_{f3}^{(i)}$ 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 $\tau_{f4}^{(i)}$) 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 3 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 $T_{jreal}^{(i)}$ that is modelled as a stochastic timed Petri net with $k$ transitions and $l$ loops and sub loops may be evaluated as:

$$T_{jreal}^{(i)} = \sum_{l \in L} \Pr\{e_l^{(i)} = 1\} \left( \sum_k \Pr\{f_{lk}^{(i)} = 1\} \tau_{lk}^{(i)} \right)$$

where

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

- $f_{lk}^{(i)} = 1$ - 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:

$$M_{Depend}^{(i)}(J_{IN}^{(i)}) = \Pr\{T_{jreal}^{(i)} \leq T_{jOUT}^{(i)}\}.$$ 

Of course, the Petri net model may be very useful in a creation the maintenance policy system of a services network, too.

8. Conclusion

It has given the functional - dependability model of a services network. The formal model consists of a tuple mathematical model and the Petri Nets one. The proposed Petri net model may be very useful in the synthesis process of the services net and its maintenance policy system.

Of course there are a lot problems with building the Petri net model of the real services net in which exist a big number of services and technical resources that are mapped to many concurrent realized tasks. A lot of possible maintenance politics, which are network reactions on hypothetic or real faults and threats, very complicate the proposed models, especially if costs of maintenance rules are considered. Additional problems are consequences of possibilities to realize the task on the base on different services and resources, of course with different costs.

Creation of the Petri Net models and evaluation network dependability measures require a special computer tools and they may be realized with using simulation techniques, based on hierarchical or object-oriented system definitions [7], [12].

Acknowledgements

Work reported in this paper was sponsored by a grant No. 4 T12C 058 30, (years: 2006-2009) from the Polish Committee for Scientific Research (KBN).

References


