Methods and software for estimation of information system dependability

Nikola Šamec *, Alen Jakupović **
* National Protection and Rescue Directorate / National Protection and Rescue Information and Communication System, Zagreb, Croatia
** Polytechnic of Rijeka, Business Department, Rijeka, Croatia
nikola.samec@duzs.hr
alen.jakupovic@veleri.hr

Abstract - Information system dependability reflects the user's degree of confidence in the system. Today's ICT allows connections between information systems in different applications and in different geographic locations. These information systems are called dependable because connections between them cause certain events in one information system to be reflected in the other. Therefore, it is important to have a method for the estimation of information system dependability that will be used in the information system development phase. This estimation will provide arguments for a possible modification of information systems before their final commissioning. The purpose of this paper is to provide an overview of the methods and software for the estimation of information system dependability. Five qualitative and two quantitative methods for estimation of information system dependability are presented. A list of ten estimation software it has been made and five of them are presented in detail.

I. INTRODUCTION

The protection and survival of complex information systems that provide services for the infrastructure of advanced society has become a national and world-wide concern of the highest priority. [30]

Real-time systems are used in the critical applications such as space flight, nuclear technology, aviation, etc. Failure of such systems can lead to major damage, loss of life and environmental impact. The common denominator of these types of failure in organizations is the loss of value in the capital market. Research in [3] states that a failure of information systems can cause a 2% decrease share values for a period of two days.

On the basis of functions and scope of the failure of the system in real time, there are three types of systems:

The first is *Safety-critical* systems required to ensure the safety of EUC (equipment under control), people and environment. Examples of such systems are: shutdown of the nuclear reactor, digital aircraft flight control, etc. [1]

A second type is *Mission-critical* systems whose failure results in failure of the mission. For example, control and coding unit (CCU) of an avionic navigation system or spacecraft, etc. [1]

The third type is *Economically-critical* systems whose failure results in the unavailability of EUC (equipment under control) or a failure such as a server or communications equipment in a business environment (system control reactor nuclear power plant, unavailability of servers in the banking sector due to a fault on the router etc.). [1]

The purpose of this paper is to present methods and software for estimation of information system dependability and to present short review of several important concepts relevant to the understanding of term dependability. Such concepts to be reviewed are the concept of dependability and terminological structure that is published by Jean-Claude Laprie in his work "Dependability: Basic Concepts and Terminology". [15] Other concepts are methods for estimation dependability with emphasis on Markov chains, methods for estimation reliability, availability, maintainability and safety, and software to support these estimation methods.

II. INFORMATION SYSTEM DEPENDABILITY

Dependability of an information system is the ability to deliver service that can justifiably be trusted. [5] There is also an alternative definition by which dependability is defined as the ability of the system to avoid frequent and severe inaccessibility of services than what is acceptable.

Dependability studies two or more systems which interact in a dependent relationship, or how an event in one system affects the operation of other systems and the overall system. Evolution of the concept of dependability through four decades is resulted on terminology that is presented by Jean-Claude Laprie.[15]

The three basic elements of observation are: attributes, threats and methods for achieving dependability.

A. Attributes

Information system dependability is a complex information system characteristic which consists of the following attributes:

- *Availability* is the system's ability to deliver the correct service. [5], [4], [2], [14]
- *Reliability* is the system's ability to continually deliver the correct service. [5], [4], [2], [14]

- *Safety* is the system's ability to operate without catastrophic consequences on the user and environment. [5], [12], [7], [2], [14]
- *Confidentiality* is the system's ability to not permit unauthorized disclosure of information. [5], [4], [2], [14]
- *Integrity* is the system's ability to prevent improper modification of the system. [5], [4], [14]
- *Maintainability* is the system's ability to be easily repaired and maintained. [5], [4], [2], [14]
- *Security* is the system's ability to protect itself from accidental or deliberate attack (is formed by combining the attributes of availability, confidentiality and integrity). [19], [2], [14]

B. Threats

Threats of the information system dependability are:

- *Fault* is improper deviation of at least one characteristics of the system. [21], [14]
- *Error* is the state of the system that can cause *failure*. Errors can be latent or detected [5], [4], [14]
- *Failure* is the permanent cessation of the system's ability to perform a required function. [21], [14]

C. Methods for achieving dependability

There are four basic methods for achieving dependability:

- *Fault prevention* is to prevent the occurrence of a fault, and achieves the quality control during the design and production of hardware and software. [5], [4], [14]
- *Fault tolerance* refers to information systems that deliver the right service due to the existence of an active fault and have well-developed immunity to malfunction. This technique is commonly used for error detection and subsequent system recovery. [5], [4], [2], [14]
- *Fault forecasting* is performed by estimating the behaviour of the system with respect to the occurrence of a failure. It is estimated the current situation, future failures and the consequences of failure [5], [4], [14]
- *Fault removal* is applied in the development and working phase of information systems. The development phase includes testing, diagnosis and repair, while the working phase includes corrective and preventive maintenance. [5], [4], [14]

III. METHODS FOR ESTIMATION OF INFORMATION SYSTEM DEPENDABILITY

These methods perform estimations of the current state of the probabilities of failure and estimations of the probability of the consequences of failure. Using these methods it is possible to estimate the degree of confidence on the ability of the system to meet specific goals. These methods are used to [19]:

- compare possible solutions
- estimate level of system resistance in the work
- estimate system confidence, required resources and costs

Estimations of system dependability can be observed through two approaches:

- *Qualitative* or *ordinal approach*, wherein the identification and classification of the system failure or combination of events which cause system failure is performed [4]
- *Quantitative* or *probabilistic approach* involves estimation of the probability to what extent measured attributes meet requirements. An important component of this approach is the measurement of attribute. [4]

A. Qualitative (ordinal) estimation methods

The following methods are shown: FMECA method, Reliability block diagram (RBD), Fault tree, Attacks tree.

a) FMECA method

FMECA or Failure Modes, Effects, and Criticality Analysis is method that was intended to be used for modelling hardware, but subsequently began to be applied on software (SEEA: Software Error Effect Analysis). Originally FMECA was developed in the '40s for the U.S. military under the standard name MIL-P-1629th. [26]

In the early 60's this method was adopted by NASA to develop space programs such as Viking, Voyager, Magellan and Galileo. [20]

Soon after, the method was extended to civil aviation and auto industries.

The method is used to identify each component or function, discovering ways in which failure occurs and what are the consequences caused by the system. It is possible to make estimations of any possible case of failure and based on the weight of the consequences caused by failure to recognize and execute a prioritization. FMECA process recognizes the critical failure modes and thus enhances the formal recognition of risk for the project and provides an incentive to change the design of the system. [19]

b) Reliability block diagram (RBD)

Reliability Block Diagram executes analysis of reliability and availability of large and complex systems using block diagrams to show network connection. The method was developed by the U.S. military under the standard name MIL STD-756B.

Graphical method consists of two types of components: blocks (representing system components) and dummy nodes (for connections between the components). Graphic topology describes how the reliability of individual components affects the reliability of the system. Dependability of blocks and nodes models is dependent on the condition of its components. The system is considered operational if the ultimate dummy nodes are connected at any point of time. If it does not, the system is considered to be faulty. Serial systems are considered not redundant while parallel systems are considered redundant systems. [9]

c) Fault tree

This method was developed by H.A. Watson from Bell Laboratories in cooperation with the U.S. Air Force, and

later, this method was extended to the field of civil aviation. [8] A model presents a graphical representation of a combination of events that cause the occurrence of the event. Tree can be used for modelling of hardware and software failures, human errors, errors in system maintenance and environmental impact on the system. The model identifies the relationships between undesired events in the system and failure of subsystems that contributes to failure of the entire system. A model for estimation of the reliability is developed by "top-down" principle, and the method can be applied at each stage of system design. The method provides qualitative and quantitative estimations of the reliability of the system. Fault tree represents an acyclic graph with internal nodes (having logic gates such as AND or OR), external nodes (leaves or basic events, which are the components of the system) and edges that represent the flow of information failure in terms of Boolean entities (TRUE or FALSE, 0 or 1). The links that are connected to the edges define the operational dependence on the system components. At any point of time, the logic value of the root node determines whether the system will be operational. [9]

d) Attacks tree

Attacks tree is based on modelling with the help of graphical and mathematical methods. This method was in its original form developed for intelligence agency and for the first time the idea of logic trees threats appeared in the literature in the late 80s. Later, B. Schneier published the concept of attack tree [6].

The method of attack tree is closely related to the concept of the idea of fault tree method because it describes a set of events that can lead to system failure. This tree can be used for modelling of all possible attacks on the system, which provide a formal, methodical way to describe the security system that is based on the different types of attacks. Attacks on the system are displayed in a tree structure, where the roots are nodes of attack targets, and methods of attacking are the leaf nodes. Safety of large systems can be modelled with a set of attack trees, where the root of each tree is an attack that can significantly damage the system. In the outline view, a tree is shown with two types of nodes (logical AND or OR). Attacks tree provides a systematic description of security vulnerabilities, thus making it possible to assess the risks and make safety decisions. [9] [25]

B. Quantitative (probabilistic) estimation methods

The big disadvantage of the above-described qualitative estimation methods is assumption of stochastic independence between components in a system which has resulted in the inability of its applications in complex systems. [17]

For this purpose the quantitative (probabilistic) estimations methods are used. These methods are based on the state space e.g. Continuous Time Markov chain, Markov reward models, Petri nets, etc.

Markov chain is a discrete stochastic process that is commonly referred to as the state transition diagram. Markov "law" says that the next step depends only on the current situation, which can be represented by the following equation (1):

$$P(X_{n+1}|X_1, X_2, ..., X_n) = P(X_{n+1}|X_n) \quad (1)$$

Markov chains know the following space states: initial, transitional and final.

Continuous time Markov Chain (CTMC) is a mathematical model that allows the state change in any moment in time. Transition matrix (describe transitions between the state) expresses transition rates instead of probabilities. Description of the condition of continuous time Markov chain can be explicitly used to monitor the status of components and subsystems for observed system. Each state represents particular error state, while transition represents component failure rate. States show the number of failures of components over time. CTMC is time homogeneous process in the way that events such as failures and repairs are independent of each other. [10]

Discrete time Markov chain (DTMC) is a mathematical model that executes a change of state of the system after a specified time. The transition to the next state depends on the transition probabilities in the transition matrix. Each row of transition matrix represents an output from the state, and each column represents the input to the state. This model is used to predict the probabilities of hardware failure. [10]

Petri Nets is a graphical form of the formal logical description of the interaction between the components or the flow of activities in a complex system. The original Petri Nets have no time dimension. For the study of dependability is necessary to introduce the duration of events associated with transient conditions. Petri Nets can be extended with the transition duration which results in depicting temporal dimension. A special case of time Petri Nets are Stochastic Petri Nets (SPN) where the trigger time is considered random variables. SPN can be automatically converted to the basic Markov model. SPN in graphic terms consists of two types of nodes - places and transitions. Places typically represent conditions in the system while transitions represent events that cause changes in system conditions. Tokens are points associated with the place and represent the status of the place. Arcs connect places and transitions. The places from which an arc runs to a transition are called the input places of the transition. The places to which arcs run from a transition are called the output places of the transition. [16]

a) Markov chains in estimation of information system dependability

Markov chains are used to estimate the availability and reliability of complex hardware systems. This probabilistic method is applied also for modelling software reliability. Approach is suitable for predicting the reliability already at the design phase of the information system, even before a "black boxes" models and real components or software subsystems become available.

Markov models can serve as a basis for Markov Reward Models (MRM), which is used to estimate the reliability of the system. It allows "rewarding" the system for time spent in states that represents readiness of the system. [13]

b) Stochastic Petri Nets in estimation of information system dependability

Stochastic Petri Nets (SPM) provides a valuable tool for describing and analyzing the system. Since the beginning of its application, this method is used to solve problems in the estimations of reliability, availability, performance and analysis of software and hardware systems. [17]

In [28] is described Stochastic Activity Networks (SAN) that is based on Stochastic Petri Nets. This method is used to estimate reliability and availability. The main concept is activities (transition in Petri nets) that have impact to the system's ability to perform. Estimation is based on the analysis of which activities are enabled in a certain state of the system, and what is the next state of the system upon completion of activities.

IV. SOFTWARE FOR ESTIMATION OF INFORMATION SYSTEM DEPENDABILITY

There are several software packages for modelling dependability. Some of them are: SURF-2, GREAT-SPN, Ultra SAN, Möbius, Sharp, DrawNet ++, SPNP, Deem, TimeNeto, DSPNexpress, adviser, ARIES, CARE III, METFAC, SAVE, SURE, ASSIST, HARP, etc. Table 1 shows some software for modelling and estimation of dependability.

A. SURF-2

Surf 2 is a tool for estimation of hardware and software systems dependability. It is based on a strict construction, validation and numerical solving of Markov models. The software was developed in the laboratory LAAS, France 1996. The system behaviour is modelled with Markov chains and Stochastic Petri Nets. The main idea of the software is to have a simple method that compares the reliability of various system architectures. [24]

B. Great-SPN

GreatSPN is a tool that supports the design and the qualitative and quantitative analysis of Stochastic Petri Nets and Stochastic Well-Formed Networks (SWN). GreatSPN appeared in the late 80s of the last century. Since then, several different versions of this tool have been developed.

GreatSPN2.0 package consists of different software that collaborates in the construction and analysis of PN models. Using the capabilities of communication networks, it is possible to perform various analyzes on different machines in a distributed computing environment. GreatSPN2.0 modular structure enables the addition of new modules for the analysis, as well as new research results. All modules are written in the C programming language, which ensures portability and efficiency on various UNIX machines. [11], [22]

TABLE I. SOME SOFTWARE FOR ESTIMATION OF INFORMATION SYSTEM DEPENDABILITY [19]

Software name	Used methods	Designer
SURF-2	GSPN, Markov chain	LAAS, France
Great-SPN	GSPN, SWN	Torino, Italy
UltraSAN	SAN	UIUC, USA
Möbius	SAN, Markov chain …	UIUC, USA
SHARPE	Fault Tree, Markov chain, SAN…	DUKE,SAD
SPNP	SPN, SRPN, non Markov chain	DUKE,SAD
DRAWNET++	Fault Tree, SWN	U. del Piemonte orientale, U. Torino, U. Napoli, Italy
DEEM	SPN,	UNIFI-PISA, Italy
Time NET	non SPN	Hamburg, Germany
DSPNexpress	Deterministic and stochastic Petri nets	Dortmund, Germany

C. UltraSAN

UltraSAN is a software tool used for evaluation of the system based on the Stochastic Active Networks (SAN). SAN has the characteristics of Stochastic Petri Nets and queuing models. Using a variety of analytical and simulation modules, it is possible to determine following: efficiency, maintainability and feasibility. UltaSAN enables the graphical presentation of the results. To determine the valid model, it is necessary to specify a set of subnets using SAN editor and classify them according to the hierarchy. UltraSAN offers six analytical techniques for solving transient and steady state. [29]

Three techniques that allow solving the steady state are: direct steady-state, iterative steady state, and iterative deterministic steady state. To resolve transients state, transient instant of time method, PDF interval-of-time and expected interval-of-time techniques is used. [27]

D. Möbius

Möbius (Model-Based Environment for Validation of System Reliability, Availability, Performance and Security) is a software tool for modelling behaviour of complex systems. The tool was designed at the University of Illinois, USA, and was originally designed to study the reliability, availability and efficiency of computer and network systems. Its flexible approach allows engineers and scientists to present their systems in a language model appropriate to their problem area, and then accurately and efficiently solve systems using solving techniques that best suit the size and complexity of the problem. The tool supports Stochastic Petri Nets, Markov chains and Stochastic Process algebra. Models are presented numerically and graphically, and are made with the right level of detail, and have the possibility to adapt to the behaviour of the system of interest. This tool can build detailed mathematical expressions that measure the correct information about the system (e.g., reliability, availability, performance and security). Measurement can be carried out at certain points of time (some time period or time when the system reaches a steady state). [18]

E. SHARPE

SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) is a tool that provides a specification and solution methods for the most commonly used type of estimation model for performance, reliability and feasibility. Models that are present in the SHARPE are Fault Tree, Model Queuing and State Model (Markov chains, semi Markov chain with reward and Stochastic Petri Nets). For each model SHARPE has preferred analytical algorithms. This tool allows measurement of models that can be used as a parameter to another model. Therefore it is considered to be a hierarchy oriented tool. The user interface supports command-line and graphical user interface that is made in Java. SHARPE is developed at Duke University, USA. [23]

V. CONCLUSION

The term dependability is relatively new and is based on the real needs of reliable and available information system that is easily maintained, has protected confidential information, is secure from external and internal threats and is safe for environment.

First estimation methods for information system dependability are developed for the needs of mission and safety critical systems in the space programs of NASA, ESA and in the military industry. These methods are now customized for use in economically-critical systems.

The fact that the failure of information system in economically-critical systems may result in significant costs, has led to a need to estimate dependability before information system is implemented. For this purpose, there is a space for development of a new methods and appropriate software for estimation of information system dependability in economically-critical systems.

REFERENCES

[1] A. K. Verma, S. Ajit, M. Kumar, Dependability of Networked Computer-based Systems. Springer, 2011,1; pp 3-5

[2] A. Avizienis, J.C. Laprie, B. Randell, Dependability and its Threats: A Taxonomy, 18th IFIP World Computer Congress, Toulouse 2004

[3] A. Bharadwaj, M. Keil, M. Mähring, Effects of information technology failures on the market value of firms. Journal of Strategic Information Systems. 18:66-79. 2009

[4] A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr. (2004) Basic concepts and taxonomy of dependable and secure computing. IEEE Transaction Dependable and Secure Computing, January-March 2004 (vol. 1 no1.), pp. 11-33

[5] A. Avizienis, J.C. Laprie, B. Randell. Fundamental concepts of dependability. In: Proc. of 3rd Information Survivability Workshop, pp 7-11, October 2000

[6] B. Schneier. Attack trees: Modelling security threats. Dr. Dobbs journal. pp. 21-29. December 1999.

[7] C.B. Jones, B. Randell, The role of structure: a dependability perspective. In Structure for Dependability: Computer –Based Systems from an Interdisciplinary Perspective. pp. 3-15.Springer 2005

[8] C. A. Ericson II. Fault Tree Analysis-A History from the Proceedings of The 17th International System Safety Conference, 1999

[9] D. M. Nicol, W. H. Sanders, and S. K. Trivedi. Model-based evaluation: From dependability to security. IEEE Trans. on Dependability and Security, 1(1):48-65 2004.

[10] Dr. P. Tröger, Dependable Systems, State-Based Dependability Modeling, Dependable System Course 2013, pp 8-29

[11] GreatSPN, http://www.di.unito.it/~greatspn/index.html, 30.01.2014.

[12] IEC 61508: Functional safety of electric/electronic/programmable electronic safety-related systems, Parts 0-7; Oct. 1998-May (2000)

[13] I. Eusgeld, B. Fechner, F. Salfner, M. Walter, P. Limbourg and L. Zhang. Hardware Reliability. In Dependability Metrics. 9; pp. 59-103. Springer 2008

[14] A. Jakupović, Utjecaj oslonjivosti informacijskog sustava na poslovne organizacije, Zbornik Veleučilišta u Rijeci. 1 (2013) , 1; pp. 165-178

[15] J.C. Laprie, A.Avizienis, H.Kopetz. Dependability: Basic Concepts and Terminology, Springer-Verlag, 1995

[16] J. Happe, Analytical Performance Metrics. In Dependability Metrics, Springer 2008, pp214-218

[17] J. K. Muppala, R. M. Frics and K. S.Trivedi, Techniques for system dependability evaluation. In Computational Probability. pp 9-10. Springer, 2000

[18] Möbius, https://www.mobius.illinois.edu/, 30.01.2014.

[19] M. Kaâniche, K. Kanoun, J.C. Laprie, Dependability and Security evaluation of computer-based systems 2009

[20] NASA, Failure Modes, Effects and Criticality Analysis (FMECA), Practice no. PD-AP-1307. 1999

[21] R. Isermann, Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, Springer 2006

[22] S. Baarir, M. Beccuti, D. Cerotti, M. De Pierro, S. Donatelli and G. Franceschinis,The GreatSPN Tool: Recent Enhancements, ACM Performance Evaluation Review Special Issue on Tools for Performance Evalaluation, Volume 36, Issue 4, September 2009

[23] SHARPE, http://sharpe.pratt.duke.edu/, 30.01.2014.

[24] SURF-2, http://homepages.laas.fr/surf4tst/what-uk.html#modele, 30.01.2014

[25] T. R. Ingoldsby. Attack Tree-based Threat Risk Analysis. Amenaza Technologies Limited. 2010, pp. 3-9

[26] U.S. Department of Defense. Procedures for Performing a Failure Mode, Effects and Criticality Analysis. MIL–P–1629, 1949

[27] W. D. Obal II, M. A. Qureshi, D. D. Deavours, W. H. Sanders, Overview of UltraSAN, in Computer Performance and Dependability Symposium, 1996., Proceedings of IEEE International, 4-6 Sep 1996

[28] W. H. Sanders and J. F. Meyer, „A Unified Approach for Specifying Measures of Performance, Dependability, and Performability", in Dependable Computing for Critical Applications, Vol 4: of Dependable Computing and Fault-Tolerant Systems (ed., A. Avizienis and J. Laprie), Springer-Verlag, 1991

[29] W.H. Sanders, W.D.Obal II, M.A.Qureshi and F.K. Widjanarko, UltraSAN Ver3: Architecture, Features, and Implementation, roceedings of the AIAA Computing in Aerospace 10 Conference, San Antonio, TX, March 28-30, 1995, pp. 327-338

[30] Wilson, STRATUS Computer System", in Resilient Computing Systems, 1985, pages 208-231