DEPENDABLE COMPUTING SYSTEMS IN SUPPORT OF TRANSFORMATION OF THE FORCE INFORMATION INFRASTRUCTURE

Vyacheslav KHARCHENKO, Vladimir SKLYAR,
and Oleg ODARUSCHENKO

Abstract: This article analyzes the approaches related to dependable computing systems and Force Information Infrastructure creation out of undependable components (systems). Taxonomies of dependable systems and stages of dependable systems paradigms evolution are discussed. A life-cycle model and principles of development and assessment of dependable systems by means of multi-version technologies are considered. Some general tasks and methods for providing and managing dependability are analyzed. The Models, Techniques and Tools Infrastructure for supporting transformation of the Force Information Infrastructure is proposed. Warfighter Information Network (WIN) development is considered in this article as an example of the principles of using dependable computing systems implementation to support transformation of the Force Information Infrastructure.

Keywords: Force Information Infrastructure, Dependable Computing System, 3M-Conception.

Introduction

The rapid development of information technologies (IT) in all areas of application affects also the Force Information Infrastructure (FII). The modern FII includes, for example, systems of combat control and liaison, control systems of armament, telecommunication systems, databases and knowledge bases, decision support systems, etc.

FII disruption can lead to the following damage for society and economy:

- Threaten the life of humans;
- Disable government to decide and act freely;
- Impact the prosperity of economy;
- Threaten the ecology, the lives of animals, or destroy major natural cover;

- Disrupt a nation's societal, social, and political structure.

Obviously, it is of general interest to protect a nation's critical FII against a broad range of threats and of European interest to extend the same level of protection beyond national borders and across Europe.

Recently, dependability as a key attribute and paradigm of dependable computing has extended over the whole complex of computer-based systems. It concerns, first of all, the so called critical infrastructures and its information core, i.e. the information infrastructure (IIS). FII is a kind of critical IIS.

Dependability (safety and security) of FII essentially depends on computer-based control and information systems (CCS) dependability. The part of failures and crashes causes stipulated by software design faults permanently increases. It concerns, for example, military aerospace systems, power-supply systems and telecommunication systems. Research performed by the authors shows that about 20% of critical IIS crashes, accidents and failures happened due to CCS faults.[1] This fact demands urgency in development and implementation of novel principles and approaches for such FII.

To provide dependability of computing systems, developers should use different approaches and techniques. One of the most important approaches to creating dependable computing systems (DCS) is multi-version or diversity implementation. Multi-version systems encompass different kinds of redundant fault-tolerant (intrusion-tolerant) systems. The applications of the multi-version approach for providing dependability and its attributes have been analyzed in the work of Strigini and Littlewood.[2]

In this article, the principles of DCS and FII development out of undependable components are considered. The main objective of the article is analysis of DCS use for the support of FII transformation in the new conditions of IT development. To this end, the following issues are discussed:

- Taxonomies of dependable systems;
- Aspects and stages of DCS and FII paradigms;
- General tasks and principles of providing and managing dependability by means of multi-version technologies;
- A lifecycle model and features of development and assessment of dependable systems;
- The Models, Techniques and Tools Infrastructure (MTTI) for supporting transformation of FII;
- An example of FII development.

Taxonomy of Dependability

The basic concepts of this taxonomy completely correspond to the taxonomy proposed in a key publication by Avizienis, Laprie, Randell, and Landwehr.[3] Some additions and comments are given by Kharchenko, Sklyar, and Volkovoy.[4]

Dependability of a computing system (and FII) is the ability to deliver required service (set of services – megaservice) that can justifiably be trusted.

A taxonomy scheme of dependability consists of the following elements: threats (defects or vulnerabilities, faults, errors, failures), attributes that form dependability as complex property, and fault- and intrusion-tolerance (as mechanism of supporting dependability attributes).

There are three kinds of faults: physical faults (of hardware), design faults (first of all, of software), and interaction faults caused by environment and human factors.[5]

Dependability integrates the following attributes [6]:

- *Reliability* – continuity of correct service;
- *Availability* – readiness for correct service;
- *Maintainability* – ability to undergo modifications and repairs;
- *High Confidence* – ability of correct estimation of service's quality, i.e. definition of trust level to the service;
- *Safety* – absence of catastrophic consequences for the user(s) and the environment;
- *Survivability* – ability to minimize loss of quality and to keep capacity of fulfilled functions under failures caused by internal and external reasons;
- *Integrity* – absence of improper system alternations;
- *Confidentiality* – absence of unauthorized disclosure of information.

The listed attributes, which compose dependability, are primary and can be related to secondary attributes. For example, maintainability is related to repairability, checkability, and so on. It is important to note that a secondary attribute can be common to different primary attributes.

For FII, there are additional features, for example, disasterability that is secondary attribute for survivability.

Fault-tolerance is a basic means for providing dependability. This means is based on the implementation of all or some of the following operations (see Figure 1 and Table 1): F1 – fault-forecasting (*Ff*), F2 – fault-prevention (*Fp*), F3 – fault-detection (*Fd*), F4 – fault-diagnosis (*Fi*), F5 – fault-tolerating (*Ft*), i.e. in the sense of masking

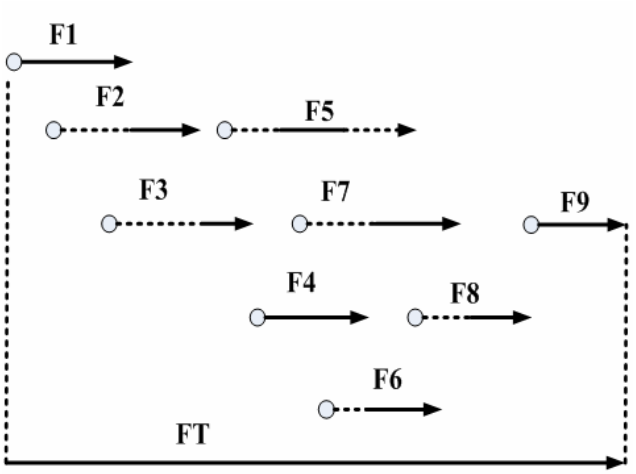

Figure 1: A Set of Fault-Tolerance Operations.

or parrying faults, F6 – fault-isolation (*Fs*), F7 – reconfiguration (*Fr*), F8 – recovery (*Fc*), F9 – restarting (*Fr*). Any fault-tolerant technique can be characterized by presence of features of elements from a set:

$$MF = \{Ff, Fp, Fd, Fi, Ft, Fs, Fr, Fc, Fr\}$$

The above-mentioned taxonomy is characterized by the following (according to Kharchenko, Sklyar, and Volkovoy) [7]:

Table 1: Correspondence between Attributes of Dependability and Fault-tolerance Operations.

Attributes of dependability	***Fault-tolerance operations and supported attributes (signed +)***								
	F1 (*Ff*)	F2 (*Fp*)	F3 (*Fd*)	F4 (*Fi*)	F5 (*Ft*)	F6 (*Fs*)	F7 (*Fr*)	F8 (*Fc*)	F9 (*Fr*)
Reliability			+		+				
Availability			+	+		+			+
Repairability		+	+	+		+	+	+	
Maintainability	+	+	+	+			+		
High confidence			+						
Safety	+	+	+		+	+	+		
Survivability	+	+	+	+	+	+	+	+	+
Integrity			+		+				
Confidentiality			+		+				

- In the definition of dependability the term "required" is added, which emphasizes the specificity of service and eliminates ambiguity;
- "Confidentiality" and "survivability" can be components of "dependability" according to purpose and specific usage of a particular computing system (e.g., the necessity of access restriction to service information have to be reflected in the confidentiality property); "high confidence" is also a primary attribute because this is represented in the definition as "justifiably be trusted;"
- "Dependability" and "security" have common attributes (namely "integrity" and "confidentiality") and specific attributes (e.g., for security it is "authenticity"). Therefore, the use of the expression "dependability and security" is not sufficiently correct;
- "Fault-tolerance" is not an attribute of "dependability," but it is defined as a means for maintenance of dependability's attributes. In our opinion, such interpretation is, first, defensible as far as means like that can provide or enhance different attributes and, second, it should increase the attention to the "intrusion-tolerance" aspect.

Dependable Systems Paradigms: Aspects and Stages of Evolution

The evolutionary analysis of development and ensuring the problem of computing systems dependability is carried out considering the following aspects:

- *Changing of elements* the systems are synthesized (produced) from (what do the systems consist of);
- *Changing of systems* as an object of synthesis and a dependability-guarantee feature (what is being synthesized);
- *Changing of elements and system properties*, which are considered and must be guaranteed (why, for which dependability properties, taking into consideration the evolution of dependability concept, are these or those methods or means used);
- *Changing of basic principles*, *methods and means* that provide the required system properties (how is the system dependability being provided).

Thus, evolution is realized in a multi-dimensional space: elements – elements properties – systems – system properties – methods and means.

Reliable System out of Unreliable Components

An initial paradigm is formulated in 1950s by John Von Neumann [8] as "reliable system out of unreliable components" (RS/URE).

For this case, the system is relay and digital devices and the main solution method of the problem is a parallel reservation of the simplest radio and electronic elements without using special automatic recovery means or a majority reservation without means of checking (passive fault-tolerance).

For this stage of evolution only one type of redundancy was representative – it was a structural redundancy.

Fault-tolerant Devices and Systems out of Unreliable (Hardware) Components

In the 1960s and 1970s, fault-tolerance became dynamical attribute introducing on-line testing and reconfiguration [9] (the paradigm was "synthesis of fault-tolerant devices and systems out of unreliable hardware components").

Elements of systems at this stage were chips of small, middle, and large integration level; hence, systems were complex digital devices, computers, and computing systems.

The main methods of the second stage were different kinds of reservation, such as structural, informational, and temporal reservation, and special means of checking, diagnosis, reconfiguration, and recovery of computational and control processes.

Fault-tolerant and Secure Systems out of Unreliable and Insecure (Software and Hardware) Components

The 1980s and 1990s then were concerned with the development of the paradigm "synthesis of fault-tolerant systems out of unreliable software and hardware components."

This paradigm reflected the growth of the weight of software as factor of unreliability. During that time, the concept of multi-version design and implementation of computing system was developed.

The multi-version approach aimed at neutralization of both physical faults and design faults by means of different kinds of redundancy and diversity. Simultaneously, another paradigm had been considered – "synthesis of reliable and secure systems from unreliable and insecure components."[10]

Dependable Systems out of Undependable Components

The elaboration of Internet technologies and service-oriented systems (at the end of 20th and at the beginning of 21st century) has resulted in the emergence of the paradigm "synthesis of dependable systems out of undependable components" (DS/UDC).

At this stage, components of distributed and global systems are computers, cluster subsystems, Web-components. The multi-version concept, which was used earlier only as a methodology for reducing design faults, has been expanded and united in the triad "multi-version systems – technologies – projects."[11]

Dependability attributes depend on both system components' (products) properties and development processes' properties, so the multi-version approach is aimed at all aspects.

IIS Dependability Paradigm

We can now assert that the implementation of the DS/UDC paradigm is extremely important for IIS, including FII. Taking into consideration that FII consist of a lot of computing systems and networks, this paradigm should be formulated for them as "dependable infrastructures out of undependable systems." Therefore, the next sub-stage of stage DS/UDS is devoted to the development and implementation of this paradigm.

Tasks and Principles of Providing and Managing Dependability

Development of Dependable Web-services

The service-oriented architecture is an example of IIS. The Web-services have to work under conditions of hardware failures (effects of operational and developmental hardware faults), software failures (effects of software design faults), and failures caused by attacks or accidents (interaction faults).

For development of Web-services from multi-version Commercial-Off-The-Shelf (COTS) components, the task of analysis is as follows: A service-oriented system W joins a set of specific services W_k, $k = 1,\ldots,K$, where each service W_k has a subset of dependability properties G_k, so there is a need to find system dependability properties GW.

The task of synthesis is as follows. There is a set of services:

$$MW = \{W_r\},\ r = 1,\ldots,R$$

with the same functionality and certain properties G_r; it is necessary to obtain a service-oriented system W^* that joins subset of services $\Delta W \in MW$, for which dependability GW is above or equal to a required G_{req} and the costs are acceptable. Restrictions to other attributes can also be imposed.

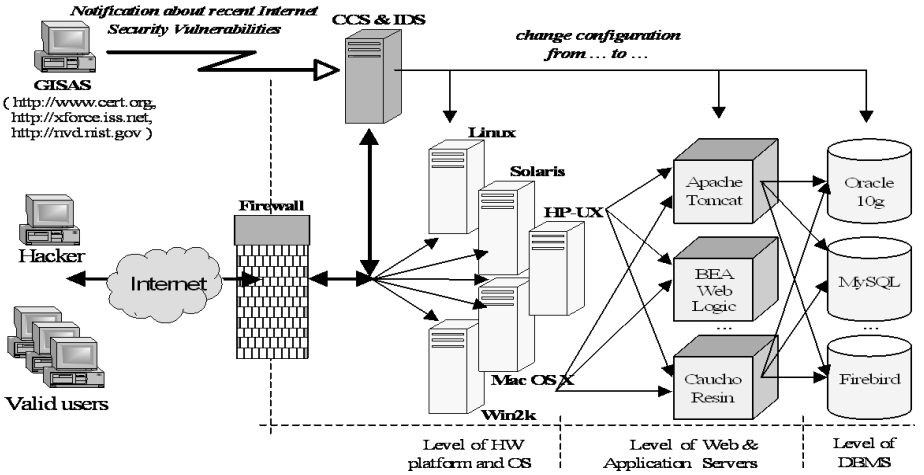

Figure 2: An Example for Dependable Multi-Version Web-System.

An example of dependable multi-version Web-system is given in Figure 2 (this architecture was proposed by Gorbenko, Kharchenko, Tarasyuk, and Furmanov).[12] Such a Web-system can be used as part of FII. This system has three levels with implemented multiversity:

1. Level of hardware platform and operating system;
2. Level of Web and application service;
3. Level of database.

Optimization of Maintenance of DCS

There are several problems related to the maximization of dependability properties in terms of amount and maintenance rate optimization, considering physical faults. A similar problem has experimental solution for a kind of computing system subject to software faults. The influence of external actions (external intrusions, attacks, accidents, etc.) provides a basis for formulation of an optimization problem for selection of a maintenance policy for dependable computing systems.

In other words, maintenance optimization considers both technical and information system's state (i.e., it is directed towards maintaining hardware and software fault-tolerance and also towards the maintenance of security by means of updating and changing protection tools and services).

Multi-version Approach and 3M Conception

The multi-version approach is based on the principle of diversity, which was originally used to reduce common mode failure possibility. It turned out that diversity also allows improving security (integrity and confidentiality), for instance, the effectiveness of two-version digital.

Diversity and the multi-version approach open up the possibilities for improving security of service-oriented systems owing to the procedure of adaptive choice of diverse Web-services' configurations, operating systems, and applications.[13]

The multi-version approach (MV) can be efficiently supplemented by a multi-parametric adaptation (MA) and a multi-level control degradation (MD).[14]

MA corresponds to organizing a few software and hardware supported contours controlling the reconfiguration, taking into consideration the modes and number of failed components.

MD is a redistribution of redundant and irredundant mobile resources and correction, if necessary, of system purposes for minimizing degradation. The MV, MA and MD approaches together form the 3M conception.

The experience gained from applying the multi-version approach and the 3M-conception for different DCS development and upgrade is analyzed in Table 2.[15] This experience, the proposed methods and techniques could be used for development and transformation of FII.

Table 2: Experience from Application of the Multi-version Approach to Different DCS.

Basic Solutions	Application Area	Implementation
Critical software development	Software of Aerospace systems, Nuclear Power Plants control systems	Tools for supporting safety review and independent verification and validation of critical software
Critical programmable logic devices based systems development	Aircraft on-board control systems, Nuclear Power Plants control systems	Anti-icing and air-conditioning systems of the AN-140 aeroplane Reactor protection systems for Ukrainian Nuclear Power Plants
STRATUS-based systems development	High-availability banking systems	In process of development
Web-services development	Dependable service-oriented systems	System INDECS for dependable Web-service upgrade and integration

Technical Information and Technical Information Organization States

The application of the multi-version technology allows managing dependability (reliability and integrity, or confidentiality), taking into consideration the real technical-information state. This state, first of all, is based on the principle of addition used in complex systems theory, i.e. reasoning from a state of external environment and occurrence of interaction faults. These faults have influence on integrity and confidentiality attributes. A technical-information-organization state takes into consideration the human factor for IIS and the risks of organization failures.

Development and Assessment of Dependable Systems

Whereas design diversity is the base for providing dependability, the development process for dependable systems has to be organized according to the multi-version approach. So, the developers have to take into account the specific character of the dependable systems life cycle (independent work of development teams on redundant versions/ channels, control of versions, overhead costs for redundancy production and N-version results estimation, etc.).

For the purpose of formalized planning and management of dependable system life cycle, the multi-version life-cycle model (MLM) has been proposed. Similarly to the classical life-cycle models, MLM defines in a formal way a list and sequence of development stages and activities and gives in addition tools and methodology to define the number of versions at each stage and the means for introducing diversity into these versions.

The multi-version lifecycle model is the part of multi-version technology, which is defined for a particular project of dependable system, and accounts for project risks, costs, and requirements. The formal definition of the multi-version lifecycle is

$$MVLC = \{S_i, G_i, U_i, m_i,\ i = 1, \ldots, n\},$$

where S_i denotes stages; G_i – operations for introducing diversity; U_i – operations for processing the obtained versions; and m_i – the number of versions produced at stage S_i.

Operations G_i and U_i are based on diversity-seeking decisions and have to be chosen for a project, taking into consideration their compatibility.

Measures of two kinds can be used for dependability assessment: vector measures, which form a set of measures for evaluation of individual dependability attributes (reliability, availability, integrity, etc.), and scalar measures, giving a generalized estimator.

The most common example of scalar measure is the probability of delivering a required service for service-oriented systems. On the assumption of faults independence, this measure can be found as product probability of fault absence (or fault tolerance). Further, these probabilities can be found out of probability values of all or essential operations from MF concerning different faults (vulnerabilities).

The scalar measure can also be obtained from a metric measure by means of hierarchical model of attributes and characteristics combined with convolution of radial-metrics diagrams.[16] For precise estimation, a detailed analysis of system states and transitions by means of Markov models and F(I)MEA-technique have to be carried out.[17]

The MTT Infrastructure for Supporting Transformation of the FII

To assess FII dependability, a set of different methods and techniques has to be used taking into consideration the infrastructure hierarchy. The FII should be confronted with the Models, Techniques, and Tools Infrastructure (MTTI) for supporting transformation. MTTI has to ensure comprehensive learning of the aspects related to the transformation of FII.

At present, the transformation of FII is determined by two main tendencies:

- Emergence of new possibilities, conditioned on the IT evolution;
- Emergence of new hazards, conditioned on the new security environment.

The following new possibilities of IT permit to increase the technical characteristics of FII:

- Perfection of electronic elements;
- Increase of the volume and functions of software;
- Widening of the Commercial-Off-The-Shelf (COTS) software and hardware decisions that can be used in the FII.

The following hazards for the FII exist in the new security environment:

- Hazards related to breaking of security, resulting in unauthorized persons gaining access to information and data;
- Hazards related to external impacts, such as mechanical, climatic, electromagnetic impacts, natural disasters, etc.;
- Hazards related to human errors;
- Hazards related to defects introduced during the process of development of computer systems.

The following principles of computer systems development are applicable for FII risks minimization:

- Use of structural, functional, informational or temporal redundancy with respect to the volume, which is minimally required and sufficient for the required functions implementation;
- Independence from failures of redundant equipment by physical and galvanic division of channels;
- Use of diverse technical decisions for achieving a specified objective;
- Monitoring, supervision and diagnosis of facilities;
- Application of organizational and technical resources for unauthorized access prevention;
- Application of facilities for protection from external extreme impacts;
- Realization of project management actions aimed at transformation of the FII: life cycle management, configuration management, verification, use of approbated components, etc.;
- Use of distributed hierarchical structures of grid-systems in the configuration of FII;
- Application of models, techniques and tools for support of decision-making frameworks.

The following models, techniques and appropriate tools can be used to support the transformation of FII:

- Reliability block-diagrams;
- Fault-tree analysis (FTA);
- Event-tree analysis;
- Failure modes, effects and criticality analysis (FMECA);
- Markov models;
- Petri net models;
- Static analysis of program code;
- Software reliability growth models.

Development of the War-fighter Information Network (WIN) to Support the Transformation of FII

The development of the War-fighter Information Network (WIN) is considered in this article as an example for the application of the DCS implementation for supporting the transformation of FII.

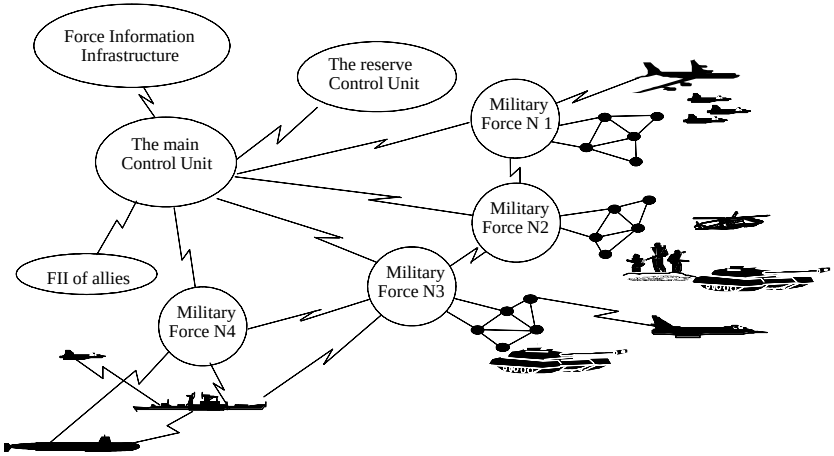

Figure 3: Structure of the Network Control System.

The main tendencies in the progress of telecommunications networks for military applications are the following:

- Enhancing the communications services spectrum in control systems (e-mail, voice mail, conferences, video-conferences, etc.) and providing the necessary level of information security and data flow capacity;
- Improvement of communications system endurance by building up of the distributed Basic Network of General Using with complex application of different communication means;
- Automation of the processes of installation, provision and renovation (troubleshooting) of communication using all types of operative switching (channels, messages, packages).

WIN is the most advanced telecommunication system for military purposes developed recently. WIN is a joint Network Control System for communication and computing, containing high-tech info-telecommunication systems and supporting all types of military requests. WIN enables the solution of the existing tasks and conforms to the FII conception.

WIN includes the following parts:

- Radio-communications system (Tactical Internet);
- Satellite communications (SatComm) system;
- Ground transport system;

- Information system;
- Network control system (see Figure 3).

DCSs are a key element of these systems and form part of FII (WIN). Interconnecting DCSs and redistributing resources taking into account the different kinds of faults (physical, design, interaction) ensure dependability of FII as a whole. In this case, we can use the paradigm "dependable infrastructures out of undependable systems."

Conclusion

Dependability is "open" feature, introducing "core-attributes" (reliability, availability, etc.) and "shell-attributes" (confidentiality, survivability, etc.), primary and secondary attributes. A model (hierarchy of attributes) of dependability is specified taking into consideration the features of the developed system. The multi-stage evolution of paradigms, methods and means of dependable computing systems has continued for the last fifty years. Here we consider the development of dependable (or dependable enough) systems out of undependable (or insufficiently dependable) components. This paradigm is part of the general approach for creation of "good" ("better") systems out of insufficiently good (or bad) components. Take as an example distributed computing ("fast" system out of "slow" computers/ subsystems). For FII, the paradigm is formulated as "dependable infrastructures out of undependable systems." We believe that the development of principles, methods and techniques according to this paradigm will be an important future research direction.

The environment where the systems operate is becoming more and more aggressive (hardware and software faults, factors of environment, intrusions, attacks, etc.). It requires a complex approach to ensure dependability as a whole and its attributes (reliability, safety, security, survivability, etc.). Therefore, it is necessary to develop and use principles and methods allowing to improve these attributes jointly, taking into consideration their complex relations as well. There are several principles of this kind (multi-version approach, multi-parametric adaptation, multi-level controlled degradation, etc.). The multi-version approach and the corresponding techniques ensure possibilities of improving main attributes of dependability introducing integrity and confidentiality. The multi-version technology based on the multi-version life cycle model allows defining a set of diversity-seeking decisions for a particular project taking into account their compatibility. To manage system dependability, techniques for operative checking, estimation and variation of technical-information and technical-information-organization states should be used.

To assess DCS and FII dependability, it is necessary to use a lot of different methods and techniques (FTA, FMEA, PSA, OD-diagrams, etc.). To this end, the following steps have to be carried out: analysis and systematization of models, techniques and

tools, which can be included into the MTTI; development of a structure of the integrated MTTI, which is intended to support the decision-making framework for creation and transformation of the FII; solving the task of selection of the optimal approach to transformation of the FII in terms of the "assessment completeness/ cost" criterion.

Notes:

[1] Vyacheslav Kharchenko, Michael Yastrebenetsky, and Vladimir Sklyar, "The Technique and the Experience of Expertise of Software for NPP Instrumentation and Control Systems" (paper presented at the 7th International Conference on Probabilistic Safety Assessment and Management, Berlin, June 2004), 2096–2101.

[2] Lorenzo Strigini and Bev Littlewood, "A Discussion of Practices for Enhancing Diversity in Software Designs," Technical Report LS_DI_TR_04 (London: Centre for Software Reliability, 2000), p. 51.

[3] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl E. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing* 1, no. 1 (January-March 2004): 11–33.

[4] Vyacheslav Kharchenko, Vladimir Sklyar, and Andriy Volkovoy, "Multi-version Information Technologies and Development of Dependable Systems out of Undependable Components" (paper presented at the Dependable Computer Systems DepCoS-RELCOMEX Conference, Szklarska Poreba, Poland, June 2007), 43–50.

[5] Avizienis, Laprie, Randell, and Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing."

[6] Avizienis, Laprie, Randell, and Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing;" Kharchenko, Sklyar, and Volkovoy, "Multi-version Information Technologies and Development of Dependable Systems out of Undependable Components."

[7] Kharchenko, Sklyar, and Volkovoy, "Multi-version Information Technologies and Development of Dependable Systems out of Undependable Components."

[8] John Von Neumann, "Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components," in *Automata Studies*, ed. Clod Shannon and John McCarthy (Princeton: Princeton University Press, 1956), 43–98.

[9] Algirdas Avizienis and Jean-Claude Laprie, "Dependable Computing: From Concepts to Design Diversity," *Proceeding of the IEEE* 74, no. 5 (May 1986): 629–638.

[10] John E. Dobson and Brian Randell, "Building Reliable Secure Computing Systems out of Unreliable Insecure Components," in *Proceedings of the 1986 IEEE Symposium on Security and Privacy* (Oakland, California, USA, April 1986, IEEE Computer Society Press), 187–193.

[11] Anatoly Gorbenko, Vyacheslav Kharchenko, Peter Popov, Alexander Romanovsky, and Artem Boyarchuk, "Development of Dependable Web Services out of Undependable Web Components," Technical Report CS-TR (Newcastle: School of Computing Science, 2004), p. 45.

[12] Anatoly Gorbenko, Vyacheslav Kharchenko, Olga Tarasyuk, and Alexey Furmanov, "F(I)MEA-Technique of Web-services Analysis and Dependability Ensuring", in *Rigorous Development of Complex Fault-Tolerant Systems*, ed. Michael Butler, Cliff Jones, Alexander Romanovsky, and Elena Trubitsyna (Md.: Springer, 2006), 153–168.

[13] Gorbenko, Kharchenko, Tarasyuk, and Furmanov, "F(I)MEA-Technique of Web-services Analysis and Dependability Ensuring."

[14] Vyacheslav Kharchenko, "The Probabilistic Assessment of Survivability and Safety of an Unmanned Control Systems with Multistage Degradation by Use of QD-diagrams" (paper presented at the 5th International Conference on Probabilistic Safety Assessment and Management, Osaka, Japan, November 2000), 525–531.

[15] Kharchenko, Yastrebenetsky, and Sklyar, "The Technique and the Experience of Expertise of Software for NPP Instrumentation and Control Systems;" Kharchenko, Sklyar, and Volkovoy, "Multi-version Information Technologies and Development of Dependable Systems out of Undependable Components."

[16] Vyacheslav Kharchenko, Boris Konorev, Olga Tarasyuk, and Andrey Volkoviy, "Techniques and Tools of Safety-Related Software Requirements Profiling and Assessment" (paper presented at the Advanced Computer Systems and Networks Conference, Lviv, Ukraine, September 2003), 95–97.

[17] Vyacheslav Kharchenko, "Methods of an Estimation of the Multi-version Safety Systems" (paper presented at the 17th International System Safety Conference, Orlando, FL, August 1999), 347–352.

VYACHESLAV KHARCHENKO is Professor, head of the Computer Systems and Networks Department, National Aerospace University "Kharkiv Aviation Institute" (Ukraine), as well as an invited researcher at the Ukrainian State Scientific Technical Center on Nuclear and Radiation Safety and leading expert of the Certification Center of Automatic Control Systems (Ukraine). His research interests are in the field of dependability and safety of critical computer systems and information infrastructures. He graduated from the Kharkiv High Military Engineering Commanding School (Kharkiv, Ukraine) in 1974, received a PhD degree in control systems from the Military Academy of Rocket Troops (Moscow, 1981) and a Doctor of Sciences degree in computer systems and complexes from Kharkiv Military University (Ukraine, 1995). Professor V. Kharchenko was Chair of the Steering and Program Committees of the International Conference on Dependable Systems, Services, and Technologies (DESSERT 2006, 2007).

VLADIMIR SKLYAR is senior researcher in the Department of Safety Analysis of Nuclear Power Plants' Instrumentation and Control Systems, Ukrainian State Scientific Technical Center on Nuclear and Radiation Safety, as well as assistant professor in the Computer Systems and Networks Department, National Aerospace University "Kharkiv Aviation Institute" (Ukraine). His research interests are in the area of safety and dependability of critical computer systems. He graduated from the Kharkiv High Military Engineering Commanding School (Ukraine) in 1992 and received a PhD degree in computer control systems from the Kharkiv Military University (Ukraine, 2001).

OLEG ODARUSCHENKO is head of the Telecommunication Systems and Networks Department, Poltava Military Communication Institute (Ukraine). His research interests are the field in dependability and survivability of telecommunication and computer systems and networks. He graduated from the Kharkiv High Engineering Commanding School (Ukraine) in 1989 and received a PhD degree in computer control systems from the Kharkiv Military University (Ukraine, 1998).