UDC 004.052.42

B.M. KONOREV[1], V.V. SERGIIENKO[1], G.N. ZHOLTKEVYCH[2], G.N. CHERTKOV[1], Y.G. ALEXEEV[1]

[1] *I&C System Certification Center, Kharkiv, Ukraine*
[2] *V.N. Karazin Kharkiv National University, Kharkiv, Ukraine*

CONCEPT OF CRITICAL SOFTWARE INDEPENDENT VERIFICATION BASED ON INVARIANT-ORIENTED MODEL-CHECKING APPROACH

Concept of critical software independent verification based on invariants (software properties invariable during the life cycle) measurement on the platform of the source software text static analysis is presented. The use of a model-checking approach (verification of software models, oriented on the measurement of invariants) along with the experimental calibration of sensitivity and pairwise the diversity degree of invariant measurement methods, allows essentially increasing reliability of the results of critical software independent verification. Proposed approach can be used for forecasting of critical software latent faults probability and assessment of testing coverage completeness.

Keywords*: software quality, independent verification, invariant, model-checking, calibration, latent faults.*

Introduction

There is urgent task for I&C (Instrumentation and Control) systems: necessity for faultless of software which realize critical functions of systems (functions, which failure leads to essential losses, including health hazard). Requirements to quality of such software are defined by the gravity of failure consequences caused by software latent faults. Software latent faults can result in possible failures of I&C systems and affect the safety of the whole system. Therefore main requirement for critical I&C systems is the implementation of software independent verification on the basis of the technological diversity principle.

The requirement of independence of verification and validation means the implementation of diversity principle based on quantitative assessment.

Proven measured implementation of the principle of the technological diversity allows obtaining trustworthy results and reaching the cost-effective use of resources.

The capability to be proved consists in provision of objective quantitative assessment of sensitivity (probability of software fault detection) of verification methods and diversity level (conditional probability of undetected faults). The decision of this problem allows obtaining the reliable results of independent verification and validation. Therefore one of the basic results of independent verification should be forecasting of software latent faults probability and assessment of testing coverage completeness.

1. Concept

The problem of verification of software, which carrying out critical functions of for I&C systems, consists in combinatorial "explosion" of software states, and, as a result, lack of current verification means for exhaustive testing of all software states to provide required quality (dependability and safety) of critical software.

Hypothesis for solution of the problem is the following:

Software is correct if integrity of all its invariants is confirmed. Invariant – software property or attribute being steady during all software life cycle.

Methodological basis of the technology is the improved model-checking (based on models) verification [1] with use of invariant-oriented software models developed on the basis of static analysis of critical I&C systems software source codes [2], including for those performed on FPGA components.

Modified model-checking approach consists in use of invariant-oriented models (IOM) of critical software for verification and includes:

1. Forming of software source code models oriented to diversified measurement for groups of invariants:

– Semantic invariants of SW variables (physical dimension, variation interval, representation accuracy) [3];

– Control flow invariants: control flow reducibility, potential attainability and demand of operators;

© B.M. Konorev, V.V. Sergiienko, G.N. Zholtkevych, G.N. Chertkov, Y.G. Alexeev

– Use of core memory (RAM) in specific software project: memory leak, repeat memory release;

– SW control flows structure (logic of execution);

– Specific invariants of I&C systems based on FPGA components («list of sensitivity», «signal race», «latches»).

2. Experimental calibration of IOM sensitivity to software faults and degree of IOM diversity with technique of software test faults injection (test faults are selected from normative faults profile of specific project).

3. Summarized assessment of latent fault probability and test coverage completeness of software source code.

2. Methodology

Model of the consolidated software quality assessment on the base of diversified invariant measurement

For assessment of software quality the following base characteristics are generally used: functionality, reliability, efficiency, usability, portability, maintainability, quality in use (including functional safety) [4]. In the offered approach software invariants are used as the primitives (primary attributes) during selection of metrics for assessment of these characteristics.

The model of the consolidated software quality assessment, presented on Fig. 1, allows generating the base for software analysis and quality assessment in the form of superposition (association) of attributes sets

$$\bigcup_{i=1}^{3} A_i$$

of internal quality (i=1), external quality (i=2) and quality in use (i=3), which are identified taking into consideration statistical connection with the model of technological maturity of software life cycle processes.

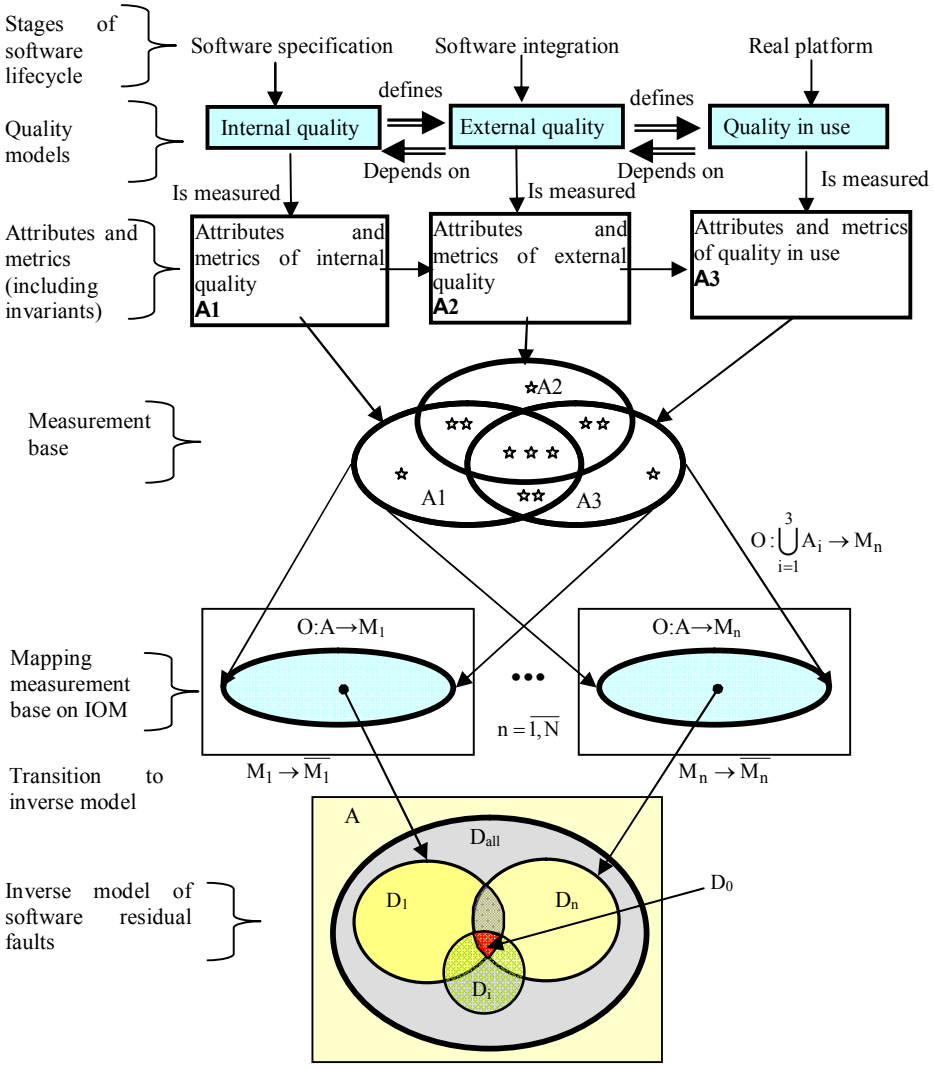

Fig. 1. Model of the consolidated software quality assessment, where A – software address field; D_{all} – software source faults; D_i – faults undetected with I method of invariant measurement; D_0 – residual faults (undetected with any of invariant measurement methods)

Possible software latent faults can cause the distortion of invariants, which results in the loss of invariance properties or software error. Latent faults appearance (software error) in use can lead to failures at system level.

Methods of measurement for various types of invariants are characterised by various sensitivity to latent software faults. On this statement the concept of diversified measurements of invariants and implementation of multiversion technology is based on a platform of static analysis for increase of reliability and precision of results of critical software independent verification. Software faults can cause invariant distortion.

Criterion for assessment of the measured software attribute - invariant is its value integrity for all cases of use.

Measurement methods of invariants are characterised generally by different sensitivity shown by probability of software fault detection. Variety degree of each pair of methods in general is presented by probability of presence of software faults, not detected by both methods. Set of diverse methods for invariant measurement, realised on platform of the static analysis of software source code, makes up the multiversion technology for diversified measurements of attributes – software invariants.

The real variety of diverse methods is a necessary condition of reliability growth (uncertainty reduction) at the use of multiversion technology of software invariant measurement.

The quantitative estimation of variety degree represents the base characteristic of method for diversified measurement of software invariants at independent verification.

Model of invariant-oriented quality assessment of I&C systems software

For the measurement of invariant values of examining projects the IOMs are forming. IOM allows controlling the integrity of invariant in the automated mode.

The mechanism of software invariant checking is presented on Fig. 2.

At the preparation stage of the static analysis the source code of checked software is brought to internal representation of the tool complex.

For this purpose syntactic and semantic analysis (in terms of compiler work) with the use of parser is performed. The received results are stored in the project database and represent the software semantic base model (BM).

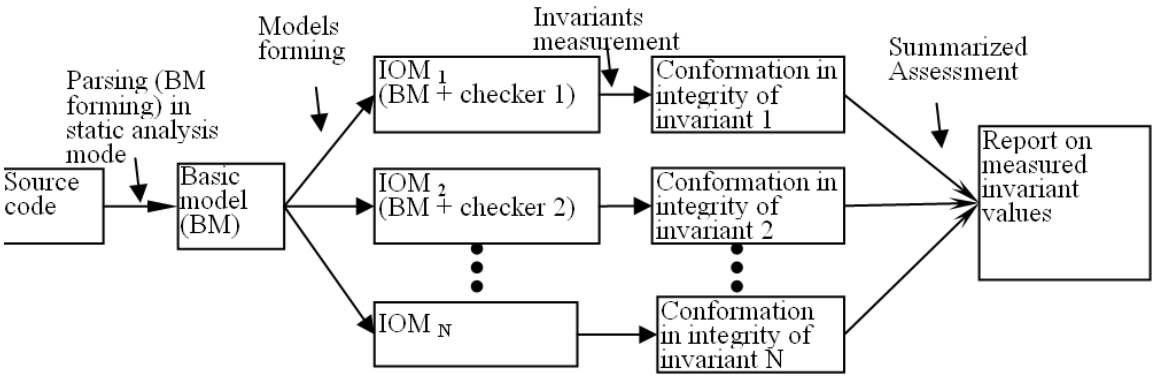

Fig. 2. Model of I&C systems software quality assessment with use of invariant-oriented models (IOM)

The invariant-oriented models are built with the use of semantic base model by elimination of information which is irrelevant to invariant.

Thus for the certain invariants dynamic interpretation of the basic model without formation of the static IOM is possible. The algorithm of integrity control was developed for each invariant.

Model of software residual and latent faults

The model demonstrates possible relative positioning of residual and latent faults sets (presented on Fig. 3).

It allows estimating benefits from use of diverse verification methods for various variants of sets allocation.

The indicator of estimation of achieved effect is the value (in %) which indicates the decrease of latent faults P_{lat} presence in the course of consequential realisation of a composition of diverse measurement methods of invariants at independent verification:

$$I = \frac{\left| D_{lat} \setminus \bigcap_{i=1}^{n} D_i \right|}{|D_{lat}|} \bullet P_{lat} .$$

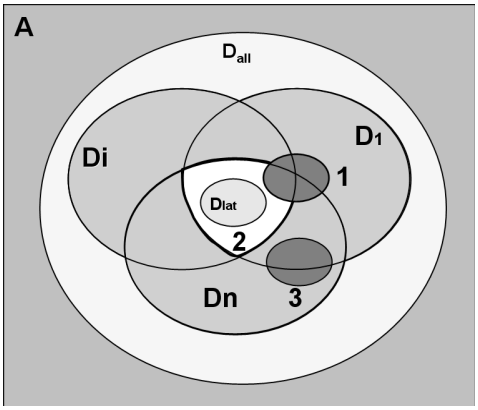

Fig. 3. Model of software residual and latent faults, where
A – software address field;
D_{all} – software source faults;
D_i –faults undetected with I method of invariant measurement; D_{lat} – latent faults

Possible variants:

a) theoretically possible case of latent fault probability decrease in 100 %

$$\left(\bigcap_{i=1}^{n} D_i\right) \cap D_{lat} = \varnothing \qquad \textbf{I=1} \text{ (position 3);}$$

b)The most adverse variant**:**

$$D_{lat} \subset \bigcap_{i=1}^{n} D_i \qquad \textbf{I=0} \text{ (position 2);}$$

c) general case:

$$\left(\bigcap_{i=1}^{n} D_i\right) \cap D_{lat} \neq \varnothing, \quad I = \overline{0,1} \qquad \text{(position 1).}$$

If latent faults (elements of set D_{lat}) will not be detected during independent verification , the subset $\bigcap_{i=1}^{n} D_i$ (even in case when benefit is I=0) can be used as boundary area of latent faults search (is a frame assessment of latent faults probability). The area where faults are absent - $\overline{\bigcap_{i=1}^{n} D_i}$. For improvement of effort assessment it is necessary to concentrate the efforts on the analysis of area $\bigcap_{i=1}^{n} D_i$. The value of the relation $\left|\bigcap_{i=1}^{n} D_i\right| \Big/ \left|\bigcup_{i=1}^{n} D_i\right|$ - defines a relative benefit from use of diverse measurement methods of invariant.

3. Implementation

The scenario of the target technology of independent verification is represented by the set of interacting processes (see fig. 4), implementing three base techniques:

- Normalization of the SW project as the object of expertise;
- Measurement of the invariants and the assessment of SW quality;
- Calibration. Integrated SW assessment. Cost-effectiveness achievement.

The functional model of the scenario (see fig. 5) is developed on the basis of IDEF0 modeling methodology and implies the hierarchy of models of various levels of detailed elaboration of scenario processes.

Basic element of the scenario is the work break-down structure. The work break-down structure defines functionally completed procedure.

The full specification of scenario work break-down structures provides development and support of scenario stages at analytical, information and organizational levels (see fig. 5).

The expected effect of implementation

Implementation of the target technology gives the opportunities to provide the controlled completeness and reliability of software verification and to achieve complete checking of compliance with regulatory requirements in a formalized project profile. This ultimately reduces the risk of latent (undetected) faults, accidents and material losses.

Reduction of routine manual operations and thus the significant reduction in complexity of the implementation of different software assessment scenarios are provided.

Opportunity of mobile instrumentation complex development for independent verification during modernization and improvement of critical software on-site with no intervention (stopping) in technological processes is provided (due to the fact that independent verification is based on static analysis of software source code of I&C systems).

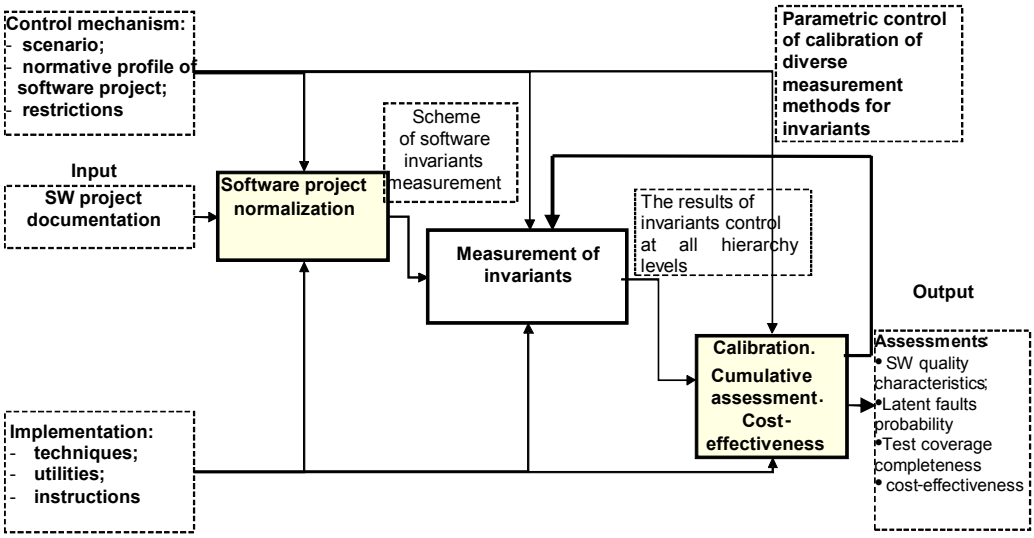

Fig. 4. Functional model of the scenario of target technology of evidential independent verification

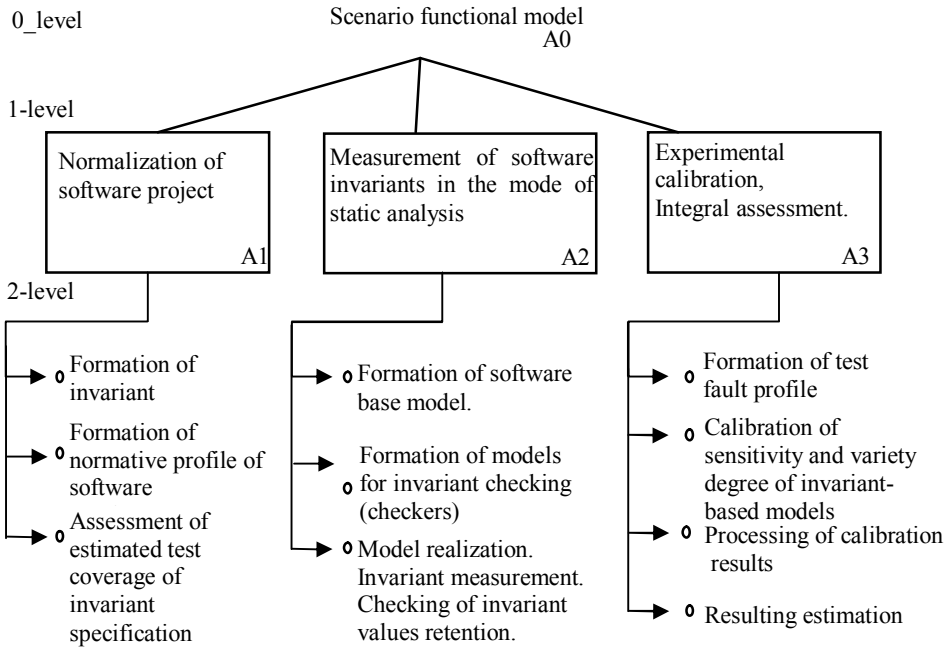

Fig. 5. Node tree for functional IDEF0-model of scenario

Conclusion

Developed target technology of independent verification and forecasting the possibility of latent faults in critical software is one of the key techniques for the analysis of criticality and evaluation of dependability and functional safety of critical I&C systems during qualification tests.

Efficiency and effectiveness of this technology mainly is determined by the real possibilities of achieving the necessary levels of dependability and functional safety of the developed I&C system.

Novelty and relevance of received results.

1. Method of evidential independent verification is developed.

This method is characterized by:

– procedures for the disclosure of the specification of invariants and the formation of invariants-oriented software models in static analysis mode of software source codes;

– procedures for experimental calibration of the sensitivity and diversity degree of invariant-oriented models of I&C system software source code, specific to a particular software project;

– multi-model diversified measurement of invariants which provides independent verification of critical I&C system software;

– evidential realization of diversity principle for achievement of reliable results;

– assessment of latent faults probability with controlled precision;

– assessment of test coverage completeness.

2. The model for assessment of latent faults probability of critical I&C system software is offered. This model differs from the known models by the use of the assessment of residual faults received by the results of calibration of the invariant-based models and provides a final assessment of latent faults probability. To achieve the required precision of the assessment during the implementation of diverse measuring methods the indicator of changes measurement of residual faults cardinal number is used.

3. The method of model-checking verification, which differs from the known methods by the use of a new class of models based on checking of invariants integrity, is improved. This method provides evidential basis to the general hypothesis of model-checking verification (correctness of models confirms the correctness of software);

4. The method of "point" injection of test software faults for the experimental calibration of the sensitivity and diversity degree of invariant-oriented models is improved. This method differs from the known models by the implemented procedure of "point" injection of test faults in accordance with the faults profile in the specific project for the specific software programming language (that allows considering the specificity of software project). For the achievement of required uncertainty degree of the results the faults injection halt criterion is used.

The offered approach (invariant-oriented model-checking verification based on static analysis of software source code) provides the following benefits:

– expanding of the real possibilities of I&C systems developers and regulatory bodies to improve the reliability and accuracy of risk prediction of I&C system abnormal functioning due to the faults in critical software in the overall context of qualification tests;

– provides evidential implementation of the technological diversity principle of critical software independent verification;

– provides an opportunity to perform the independent verification directly on-site with no intervention (breaking) in technological processes for the modernization and completion of critical software (due to static analysis mode of I&C system software source code);

– provides the possibility to assess quantitatively the limits and reduce the probability of latent faults presence in critical software;

– presents methodological basis for solving the actual problem of development of regulative and methodological and instrumental support of evaluations of dependability and functional safety of I&C systems, related to safety, in such critical areas as nuclear energy, space, transport, etc.

References

1. Карпов, Ю.Г. Model checking. Верификация параллельных и распределенных программных систем [Текст] / Ю.Г. Карпов. – СПб.: БХВ–Петербург, 2010. – 560 с.

2. Konorev, B. Qualification Testing of the Critical Software: Target Technology of Evidential Independent Verification and Forecasting of Latent Faults [Text] / B. Konorev, V. Sergiyenko, G. Chertkov // Proceedings of CrISS–DESSERT 2011; edited by V. Kharchenko, T. Tagarev. – Kharkiv: "KhAI", 2011. – V. 2. – P. 265 – 268.

3. Brukhankov, S.S. About static analysis of variables physical dimensions for critical–mission software [Text] / S.S. Brukhankov, B.M. Konorev, M.S. L'vov, G.N. Zholtkevych // Radioelectronic and computer systems. – 2010. – V. 6 (47). – P. 186 – 191.

4. ISO/IEC 9126–1:2001 Software engineering – Product quality – Part 1: Quality model.

Поступила в редакцию 1.03.2012

Рецензент: д-р техн. наук, проф. А.А. Баркалов, Зеленогурский университет, Зелена Гура, Польша.

КОНЦЕПЦІЯ НЕЗАЛЕЖНОЇ ВЕРИФІКАЦІЇ КРИТИЧНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ІНВАРІАНТО-ОРІЄНТОВАНОГО MODEL-CHECKING ПІДХОДУ

Б.М. Конорев, В.В. Сергієнко, Г.М. Жолткевич, Г.М. Чертков, Ю.Г. Алексеєв

В роботі представлена концепція незалежної верифікації критичного програмного забезпечення на основі вимірювання інваріантів (незмінних властивостей програмного забезпечення протягом життєвого циклу) в режимі статичного аналізу вихідних текстів програмного забезпечення . Використання model-checking підходу (верифікація моделей програмного забезпечення, орієнтованих на вимірювання інваріантів), а також експериментального калібрування чутливості та ступеня різноманіття методів для вимірювання інваріантів, дозволяє істотно підвищити надійність результатів незалежної верифікації критичного програмного забезпечення .

Ключові слова: якість програмного забезпечення, незалежна верифікація, інваріант, model-checking підхід, калібрування, приховані дефекти.

КОНЦЕПЦИЯ НЕЗАВИСИМОЙ ВЕРИФИКАЦИИ КРИТИЧЕСКОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ ИНВАРИАНТО-ОРИЕНТИРОВАННОГО MODEL-CHECKING ПОДХОДА

Б.М. Конорев, В.В. Сергиенко, Г.Н. Жолткевич, Г.Н. Чертков, Ю.Г. Алексеев

В работе представлена концепция независимой верификации критического программного обеспечения на основе измерения инвариантов (неизменных свойств программного обеспечения в течение жизненного цикла) в режиме статического анализа исходных текстов программного обеспечения. Использование model-checking подхода (верификация моделей программного обеспечения , ориентированных на измерение инвариантов), а также экспериментальной калибровки чувствительности и степени разнообразия методов для измерения инвариантов, позволяет существенно повысить надежность результатов независимой верификации критического программного обеспечения .

Ключевые слова: качество программного обеспечения, независимая верификация, инвариант, model-checking подход, калибровка, скрытые дефекты.

Конорев Борис Михайлович – д-р техн. наук, проф., главный научный сотрудник ХХП «СЕРТЦентр АСУ», Харьков, Украина.

Сергиенко Владимир Владимирович – руководитель испытательной лаборатории ХХП «СЕРТЦентр АСУ», Харьков, Украина. e-mail:admin@scasu.com.

Жолткевич Григорий Николаевич – д-р техн. наук, проф., зав. кафедрой теоретической и прикладной информатики, декан механико-математического факультета ХНУ им. В.Н. Каразина, Харьков, Украина.

Чертков Георгий Николаевич – директор «СЕРТЦентр АСУ», Харьков, Украина.

Алексеев Юрий Гаврилович – начальник отдела ХХП «СЕРТЦентр АСУ», Харьков, Украина.