Markov's Modeling of NPP I&C Reliability and Safety

Optimization of tool-and-technique selection

Vyacheslav Kharchenko
Department of Computer Systems and Networks
National Aerospace University "KhAI"[1]
Kharkiv, Ukraine
v.kharchenko@csn.khai.edu

Oleg Odarushchenko
Research and Production Corporation Radiy
Kirovograd, Ukraine
odarushchenko@gmail.com

Elena Odarushchenko
Poltava National Technical University named after U. Kondratiuk
Poltava, Ukraine
skifs2007@mail.ru

Valentyna Butenko[1]
v.butenko@csn.khai.edu

***Abstract* — Markov's chains are widely applied in quantitative analysis of safety-critical systems. There are few roadblocks for greater application of the Markov's chains: accounting the additional hardware and software component (or FPGA) increases the model state-space and complicates analysis; the non-numerically sophisticated user may find it difficult to decide between the variety of numerical methods and tools to determine the most accurate for their application. Obtaining the high trusted modeling results becomes a nontrivial task. We present the metric-based approach for selection of the applicable solution technique based on the analysis of several Markov chain parameters. Three optimization criteria for informed tool selection were developed to support the decision-making between the wide set of applicable software. Presented approach and criteria are applied as the stepwise tools-and-techniques selection procedure that aims to reduce the risks, increase an accuracy and optimize time needed for Markov's chains analysis. Paper presents the case study of reliability and safety assessment for industrial Nuclear Power Plant Instrumentation and Control system using the optimized tools-and-techniques selection procedure.**

Keywords— Markov chain; reliability; safety; stiffness; decomposability; sparsity; instrumentation and control system

I. INTRODUCTION

The dependability and safety assessment of the Instrumentation and Control systems (I&Cs) which are applied in the safety domain, is an essential part of the development and certification processes. The need for high accurate assessment of both safety and dependability measures is strengthen by the I&Cs application area. For instance, the I&Cs used on Nuclear Power Plants (NPP) perform the following safety-actions:

- provide to the operator an accurate and appropriate information and permit judicious action during both normal and abnormal operations;
- automatically control the main plant and many ancillary systems in different modes including emergency;
- protect the plant from the consequences of any mistakes, which the operator or the automatic control system may make;
- under abnormal conditions I&Cs provide rapid automatic action to protect both plant and the environment [1].

One of the main standards in the safety domain IEC 61508-1 provides the requirements for functional safety measures – *PFDavg* (the average probability of dangerous failure on demand) and *PFH* (probability of failure per hour) based on the system safety integrity level (SIL) [2].

The model-based evaluation is known as one of the cost-effective solutions in dependability and safety assessment of the NPP I&Cs. It is allows to capture and analyze the deficiencies such as identification of potential bottlenecks thus show the need in upgrading studies and to check whether requirements have been met during the design phase etc. Such model-based evaluation can be performed through discrete-event simulation or analytic models. Both approaches have drawbacks, since simulation can estimate results only up to a certain level of accuracy [3] and requires extra time and resources while running large models, and analytical model tend to be more abstract because of additional assumptions, which are set to make models tractable. The analytical models can be split into two groups: state-space (Markov's chains (MC), semi-Markov processes, Markov reward models, Petri nets, SAN, etc.) and combinatorial (RBD, FTA, attack trees,

ect.) models. The state-space models are preferred to non-space model, because they can easily incorporate realistic system behavior such as imperfect fault coverage, multiple failure modes, hot-swap components [4].

MC are well-know and widely applied in dependability and performability analysis of safety-critical systems, because of the ease and flexible representation of system components dependencies, synchronization and complex maintenance strategies, such as recover priority, limited recovery resource, etc. [5]. The basic property of MC is following: the knowledge of the probabilities of the system states at a given instant of time summarizes all the past and is enough to calculate how the system evolves in future can be very useful for *PFDavg* and *PFH* calculations [2].

There are few roadblocks for even greater application of the MC: accounting the additional system components exponentially increases the model state space and complicates analysis; the non-numerically sophisticated user may find it difficult to decide between the variety of numerical solution methods and tools to determine the most suitable and accurate for their application [6]. The numerical methods are limited by model size (largeness) and such very essential MC features as stiffness [7] and sparsity [8].

There are at least three important considerations for making decision between different solution techniques: efficiency and applicability of an algorithm, the structure of a matrix, size and storage needs [9]. The largeness property forces to use additional storage place, while stiffness influence on the efficiency and applicability of a numerical solution algorithm and sparsity affects the structure of a matrix. Thus obtaining the high accurate and trusted modeling results becomes a nontrivial task.

The modeling experience [10] goes in contrary with the recommendations from the one of the leading standards in the safety area IEC 61508-2010 (6th part), which asserts that "*efficient algorithms for solution of the MC were developed long time ago and implemented into software packages, so the modeler needs to focus only on the building of the model and not on the underlying mathematics*".

The previous research work [6] shows that automated selection of the solution technique based on the analysis of the main MC features can support not only the process of decision making between a wide set of approaches, but also can decrease the assessment risks.

The nowadays software market presents many software packages (SHARP, ToolKit Markov, MARCA, Möbius, ASNA, Mathematica, Matlab, Maple ect.), which can be applied during MC analysis. Each tool is limited by specific set of it internal functions, for instance different types of manual graph construction, number and accuracy of implemented numerical methods, compatibility with another packages for obtained results verification etc. Such variety of software packages and implemented functions in each of it, is extremely helpful while system modeling but poses also a difficulty when it comes to choosing the most appropriate one for a specific assessment under the need of account the MCs stiffness, largeness and sparsity properties. The main standards in safety area do not contain any additional requirements for software tool selection during application of MC for dependability and safety assessment of NPP I&Cs [6, 10].

In this paper, we introduce several optimization criteria for software packages selection, which can be applied successively to reduce the initial set of all applicable tools. We also apply the metric-based approach [27] for selection of the applicable solution technique and method, based on the analysis of MCs stiffness, largeness (decomposability, irreducibility), sparsity and fragmentedness. Using this selection procedure the modeler can also provide the verification of the earlier obtained results.

Both metric-based approach and criteria are used as the stepwise tools-and-techniques (T&T) selection procedure that aims to reduce the risks, increase an accuracy and optimize time and computational resources spend during MCs analysis.

The case study using the proposed technique was performed over the industrial NPP I&C system, manufactured by RPC Radiy. This is a three-channel FPGA-based Reactor Trip System with two parallel chassis on voting logic "1-out-of-2" in each channel. The paper presents an application of the optimized T&T selection procedure for reliability and safety analysis of RTS and procedure of results verification.

II. METRIC-BASED APPROACH FOR SELECTION OF MARKOV'S CHAINS ANALYSIS TECHNIQUE

A. Test 1: Stiffness

Stiffness is well-known undesirable property of many practical MCs as it poses a problem of finding the transient solutions. Stiffness in models is caused by [11]:

- in case of reparable systems the rates of failure and repair differ be several order of magnitude;
- fault-tolerant computer systems use redundancy, thus the rates of simultaneous failure of redundant components are typically significantly lower than failure rates of individual components;
- in models of reliability of modular software the modules' failure rates are significantly lower than the rates of passing the control from module to module.

The Cauchy problem *du/dx=F(x,u)* is said to be stiff on the interval $[x_0,X]$, if there exists an *x* from this interval for which the following condition holds:

$$s(x) = \frac{\max\limits_{i=1,n} |\mathrm{Re}(\lambda_i)|}{\min\limits_{i=1,n} |\mathrm{Re}(\lambda_i)|} >> 1, \qquad (1)$$

where the *s(x)* – denotes the stiffness index and λ_i are the eigenvalues of a Jacobian matrix ($\mathrm{Re}\,\lambda_i < 0, i = 1,2,\ldots,n$) [12]. The previous empirical work shows that quantitative value of s(x) have an impact on accuracy of different numerical methods – the higher *s(x)* value the more strict requirements imposed on the stability of chosen numerical method. Thus, we use *s(x)* as a main metric of stiffness. The *s(x)* values can be

split into three groups: high $s(x) \geq 3*10^3$, moderate $10^2<s(x)<3*10^3$ and low $s(x)<10^2$ [10].

The basic methods to overcome stiffness are described next.

1) Stiffness-avoidance approach.

The basic idea of this approach is a model transformation by identifying and eliminating the stiffness from the model, which would bring two benefits: i) a reduction of the largeness of the initial MC, and ii) efficiency in solving a non-stiff model using standard numerical methods. The approach was named an aggregation/disaggregation technique for transient solution of stiff MCs. The technique, developed by K. S. Trivedi, A. Bobbio and A. Reibmann [11], can be applied to any MC with transition rates that can be grouped into two separate sets of values – the set of slow and the set of fast states. While the transformation of the initial stiff MC brings benefits in terms of efficiency, to the best of our knowledge, no systematic study has been undertaken of the impact of the transformation (from a stiff to a non-stiff MC on the accuracy of the solution.

2) Stiffness-tolerance approach.

The main idea of this approach is using methods that are stable for solving stiff models. These methods can be split broadly into two classes: "classical" numerical methods for solution of stiff differential equations (DEs) and "modified" numerical methods used for finding a solution in special cases. Based on the analysis of earlier research works [10, 13] and conducted empirical tests for each group of *s(x)* were selected the main MC solution technique and technique for results verification [6]. Fig.1 shows normalized scale of *s(x)* with corresponding recommendations for method selection, where:

- *STA* – stiffness-tolerance approach;
- *SAA* – stiffness-avoidance approach;
- *m* – subscript, which denotes the main approach;
- *v* – subscript, which denoted the verification approach.

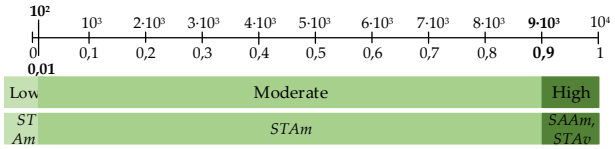

Fig. 1. Normalized scale for stiffness metric

The top values on scale (Fig. 1) are actual *s(x)* that are gained using (1) and lower symbols shows the corresponding to them normalized values on interval [0; 1]. Normalization can be performed using (2):

$$m_i = (x_i - x_i^{\max})/(x_i^{\max} - x_i^{\min}), \quad (2)$$

where m_i – normalized value, x_i – initial value, x_i^{max} - maximum on scale, for *s(x)* $x^{max} = 10^4$, x_i^{min} - minimum on scale, for *s(x)* $x^{min} = 0$.

B. Test 2: Largeness (Decomposability, Irreducibility)

MM of realistic systems are usually plagued by largeness of state space. In this case, the researched system is specified using some high-level formalisms, such as Petri nets and using this specification the underlying MC is generated. The basic solution methods for large MC are described next.

Largeness avoidance approach. The main idea of this approach is to avoid generation of the large MC from the beginning. Using largeness avoidance approach (LAA) approach the certain properties of model representation are exploited to reduce the size of the MC to obtain the measures of interest [14]. The state-level and model-level [14] lumping techniques are well-known methods of LAA approach. A state-level lumping technique is a technique that exploits the certain properties on the MC level, while the model-level lumping denotes the lumping properties on the high-level formalism and directly construct lumped MC. Another LAA technique is an aggregation, which set a condition for partition of the state space, and replacing the formed sub-sets by a single state. The aggregation in contrast to lumping gives approximate results, with or without bounds, but may result the smaller MC then a lumping technique.

Largeness tolerance approach. The largeness tolerance approach (LTA) are designed to manipulate large MC using special algorithms and data structures to reduce and store transition probabilities matrix [15]. The numerous works [15, 16] present the ideas of using binary and multi-valued decision diagrams (BDD and MDD), matrix diagrams (MD), Kronecker products, etc. to deal with state space size. The disk-based approach for steady-state and path-based approach for transient solutions are also considered in [17, 18]. Analysis of MC irreducibility and decomposability properties can help to make a prior selection between described techniques [9].

The aggregation techniques are mainly based on the decomposability approach. In this case the *degree of coupling* can be taken as measure of matrix decomposability property. For example, considering a nearly completely decomposable (NCD) MC (3), which has a matrix with non-zero elements in off-diagonal blocks are small compared with those in the diagonal blocks [19]:

$$A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix} \quad (3)$$

where $A_{11}, A_{12}, \dots, A_{nn}$ are square diagonal subblocks. The stationary distribution of π can be partitioned such as $\pi = (\pi_1, \pi_2, \dots, \pi_n)$. Assuming that $A=diag(A_{11}, A_{22}, \dots, A_{nn}) + E$, where E contains all of-diagonal blocks. The quantity (4) is referred to as degree of decomposability [19]. If E = 0 then MC is said to be completely decomposable (CD).

$$\|E\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |e_{ij}| \quad (4)$$

An irreducible MC is presented by a direct graph that is a single strongly connected component. The algorithm for determining strongly connected components is a known graph algorithm [20] Detection of such components (irreducibility property) can help in determining sub-sets for approximate aggregation technique, but it naturally applied after selecting the avoidance approach.

The value (4) can help in deciding between avoidance and tolerance approaches, thus we refer to (4) as a main largeness metric, further decomposability metric. The *E* values can be split into three groups [6]: completely decomposable (CD): $E < 0.3$; nearly completely decomposable (NCD): $0.3 \le E < 0.6$; non-decomposable (ND): $E \ge 0.6$. As in the previous test, Fig.2 presents normalized scale for *E* with recommendations for approach selection, where:

- *LTA* – largeness-tolerance approach;
- *LAA* – largeness-avoidance approach;
- *m* and *v* denote the main and verification approach, respectively, $x^{max} = 0.9$ and $x^{min} = 0$.

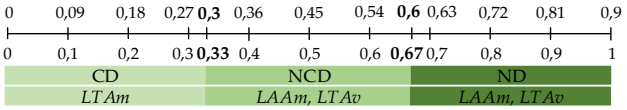

Fig.2. Normalized scale for decomposability metric

C. Text 3: Sparsity

Modeling the components interaction enlarges the MC state space significantly, thus the sparse structures are required. Transient solution methods that do not preserve sparsity are unacceptable for most large problems [8]. The direct methods for finding the steady-state solutions in case of sparse matrices may depend on the common sparse patterns, such as band/block diagonal forms, band/block tridiagonal, cyclic banded forms, etc. [21]. Paper [9] presents the formula for evaluation of the heuristic measure of sparsity – matrix score (5). It gives a measure of how the matrix elements are dispersed from the main diagonal.

Let q_i be the number of matrix elements that are a distance *i* from the diagonal. The histogram is weighted and then scaled by n^2 where *n* is matrix order. The matrix score *ms* can be evaluated using:

$$ms = \left(\sum_{i=1}^{n-1} i \cdot q_i\right) / n^2 \qquad (5)$$

In [9] authors studied the influence of *ms* value on accuracy of the tolerance techniques for MC solution, and recommended to give additional attention on *fill-in* amount for matrices with $n \ge 500$ and $ms > 0.8$. The *fill-in* is a property when initially zero matrix elements become nonzero during solution process and for which storage must be reserved.

The value *ms* can be classified into three groups [6]:

- high sparsity: $ms < 0.3$;
- moderate sparsity: $0.3 \le ms < 0.72$; l
- ow sparsity: $ms \ge 0.72$.

Fig. 3 presents normalized scale of ms with recommendations for approach selection, where $x^{max} = 0.9$ and $x^{min} = 0$.

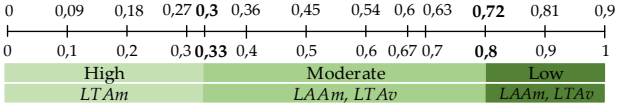

Fig.3. Normalized scale for sparsity metric

D. Test 4: Fragmentedness

The MC are widely applied to analyze the dependability of physical components of safety-critical systems. Assessing the dependability of software components depends on how well the component has been tested, is it available and whether it is a reused or new component [22]. The verification and validation (V&V) phases are strongly managed by requirements and recommendations of international standards (see standards IEC 60800-2006, IEC 61508 - 2010). There are many required procedures to test the software component, such as documentation analysis, problem review, static code analysis, etc [13]. Nevertheless, the residual software bugs can appear during system operation. In this case, to predict the general system dependability we need to observe not only hardware and software components separately, but also analyze their interconnection and total influence on system dependability.

In the previous research works [6, 13] we applied the *multi-fragmentation principle* to present such complex interconnection between system hardware and software. The main idea is to capture and represent using MC the plausible phenomenon – variation of software failure – that is well accepted on practice [10]. Using this principle the model can be divided into N_{fr} fragments that are with the same structure but may differ in one or more parameters. The number of fragments N_{fr} in MC depends on the number of expected undetected software faults n_i in *i*-different software versions:

$$N_{fr} = \prod_{i=1}^{m} (n_i + 1) \qquad (6)$$

The structure of fragment and number of such fragments can help to determine the complexity of resulting multi-fragmental model (MFM) and thus help in making decision between avoidance and tolerance approaches. In this paper, we use N_{fr} as a metric of fragmentedness. Based on N_{fr} value the MC can be classified as follows: low-fragmented: $N_{fr} < 6$; moderately fragmented: $6 \le N_{fr} < 15$; highly fragmented: $N_{fr} \ge 15$. The normalized scale ($x^{max} = 30$ and $x^{min} = 0$) with recommendations for approach selection is presented on Fig. 4.

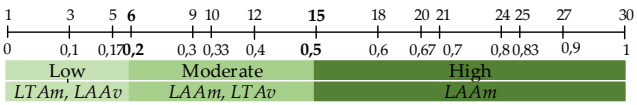

Fig.4. Normalized scale for fragmentedness metric

E. *Metric-based Diagram*

The recommendations for selecting the solution approach presented on Fig. 1 – 4 were received separately for each characteristic. However, it is important to consider each MC feature, while making decision on the most efficient technique.

In this section we present the metric-based diagram (Fig. 5), that incorporates all MC characteristics and supports the approach selection based on some specific combination of *s(x)*, *E*, *ms* and N_{fr} values. The diagram passed verification on the wide range of MCs.

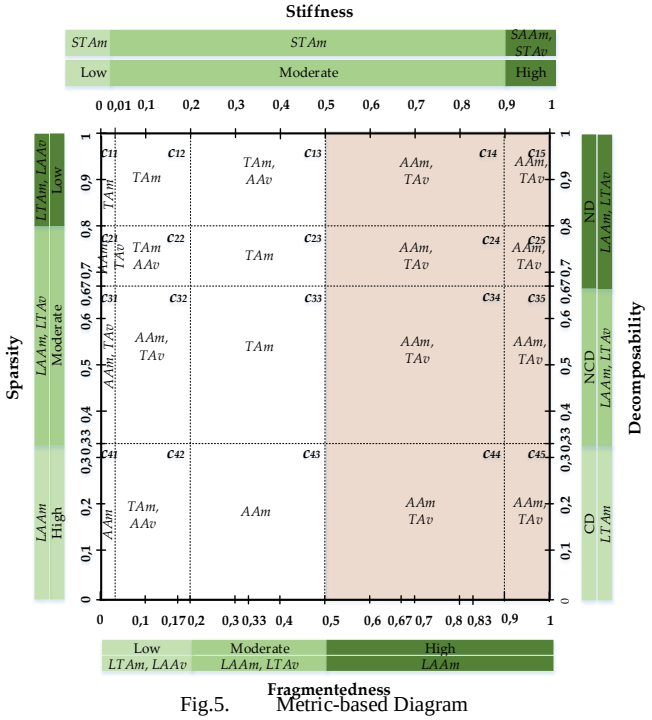

Fig.5. Metric-based Diagram

The diagram is applied in three main stages:

- Calculation of stiffness, decomposability, sparsity and fragmentedness metrics using (1), (4) – (6).
- Normalization of the received values using (Eq. 2).
- Marking of the normalized metrics values on the appropriate scale and creating the intersection by drawing perpendicular to the opposite side. As a result, we receive the rectangle, placed in one of the internal zones $c_{ij}, i \in (\overline{1,4}), j \in (\overline{1,5})$.
- Each zone provides the recommendation for selection of avoidance or tolerance approach, *AA* or *TA* respectively.
- If rectangle is placed in two or more zones, selecting the recommendations from zone that contains the larger area of the rectangle. The colored inner zones define the SAA or STA use .

III. SOFTWARE PACKAGES SELECTION CRITERIA

A. *Criteria 1: Amount of Required Functions*

At the beginning of modeling phase the researcher need to set up the initial minimal set of required functions, which selected software package is expected to support, for instance:

- type of MC construction: manually (graphically or in matrix form), or by generating from the high-level formal model. It should be note that manual type can be acceptable only for construction of MC with a small state space and cause the need in additional model verification to eliminate the typing errors.
- export (import) of underlying matrix of Kolmogorov differential equations to (from) the text or table editor. The packages, which are used during dependability and safety assessment of safety-critical systems, must be compatible to other packages in purpose of obtained results verification.
- functions for metric-based analysis. Several mathematical packages implement the mechanism of switching between numerical methods in case of detecting stiffness or sparsity. The section 2 shows the necessity of accounting every MC metric during model analysis.
- implemented numerical methods for steady-state and/or transition measures calculation.
- embedded feature of reports generation.

The problem of tool selection based on the criteria 1 can be presented using the discrete programming formalism.

The $M=\{h_1, \ldots, h_2\}$ is a set of required functions h_i, where $i=1,\ldots,N$, and $S=\{S_1,\ldots, S_n\}$ is the collection of tool sets S_i $(i=1,\ldots,n)$, which contain required functions from set *M*. Each tool S_i gets the c_j value, which is a benefit coefficient selected in an expert way. Thus, we need to find a number of subsets $S^* \subset S$ which will cover the *M* with maximal c_j value.

The $A=||a_{ij}||$ is a Boolean matrix of size $m \times n$, which elements follow the rule: $a_{ij} = 1$, if $h_i \in S_i$, $a_{ij} = 1$, if $h_i \notin S_i$.

The formal problem statement is of form, where x_j is Boolean variable that is equal to *1* if S_i is in *M* covering set and *0* in other case, thus defining the *j* package:

$$f_1(x) = \sum_{j=1}^{n} c_j x_j \to \max, \qquad (7)$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \geq 1, i = 1,\ldots,N, \\ x_j \in \{0,1\}, \; j \in N. \end{cases}$$

B. *Criteria 2: Accuracy of Implemented Numerical Methods*

The accuracy level is one of the main requirements to numerical methods embedded in the selected software package. In this case, the wide amount of numerical methods with

required accuracy level, which can be used for steady-state or transient MC analysis reflects the tool preference. Thus, the amount of numerical methods that satisfies required accuracy level can be treated as weighting coefficient of the tool. Review of Mathematica, MATLAB and Maple documentation showed the number of implemented numerical methods for steady-state and transient analysis of MC with different accuracy level – $\varepsilon_1 < 10^{-6}$ and $\varepsilon_2 \geq 10^{-6}$ (Table 1).

TABLE I. NUMBER OF IMPLEMENTED NUMERICAL METHODS FOR MC ANALYSIS

	ε	Mathematica	MATLAB	Maple
Steady-state analysis	$\varepsilon_1 < 10^{-6}$	3	2	3
	$\varepsilon_2 \geq 10^{-6}$	2	1	2
Transient analysis	$\varepsilon_1 < 10^{-6}$	9	4	7
	$\varepsilon_2 \geq 10^{-6}$	3	3	5

As in the previous case, we can use the discrete programming formalism to present the problem of tool selection based on the criteria 2:

$$f_2(x) = \sum_{j=1}^{n} b_j x_j \to \max, \qquad (8)$$
$$\begin{cases} \sum_{i=1}^{n} b_j x_j \leq C_{\max}, \\ x_j \in \{0;1\}, j \in N, \end{cases}$$

where x_j – defines the j-package, b_j – number of numerical methods in x_j package which satisfies the accuracy requirement, C_{max} – maximal number of methods, which researcher is planning to apply.

C. Criteria 2: Price of the Tool

The nowadays software packages offer a wide set of license types, starting with enterprise license up to the academia. Based on the license user obtain different amount of package functional and use different support programs. Thus, taking into account the available project budget the problem of tool selection based on price can be also presented using discrete programming formalism:

$$f_3(x) = \sum_{j=1}^{n} d_j x_j \to \min, \qquad (9)$$
$$\begin{cases} \sum_{i=1}^{n} d_j x_j \leq C_b, \\ x_j \in \{0;1\}, j \in N, \end{cases}$$

where x_j – defines the j-package, d_j – j-package price under the certain license, C_b – budget.

D. Multicriteria optimization problem statement

It should be noted that all criteria have to be considered while selecting the applicable software package. Thus, we can formulate the multi-criteria optimization problem. The number of embedded functions, methods and cost can be assumed as constant values for some time interval, we can normalize all presented local criteria 1-3:

$$R_i = \frac{f_i(x) - f_i(x)_{\min}}{f_i(x)_{\max} - f_i(x)_{\min}}, \qquad (10)$$

where $f_i(x)$ – initial value of analyzed i-criteria, $f_i(x)_{min}$ and $f_i(x)_{max}$ – minimal and maximal values of analyzed i-criteria, $i \in \{1, 2, 3\}$.

Let us set the priority coefficients of local criteria as a part of the global one (λ_i), and $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $\lambda_i > 0$, $i \in \{1, 2, 3\}$.

The scalar of the global criterion is an additive convolution of vector criteria components:

$$W(x) = \lambda_1 f_1(x) + \lambda_2 f_2(x) + \lambda_3 f_3(x) \to \max \qquad (11)$$

The well known multi-objective optimization methods [26] can be applied to obtain the (11) problem solution.

The researcher can also apply the criteria 1 – 3 successfully, depending on the criteria priority, to decrease the initial software packages set M.

IV. CASE STUDY

Here we illustrate the application of metric-based approach and tool selection criteria for dependability and safety assessment of typical NPP I&Cs. This is a Reactor Trip System constructed on the FPGA-based digital platform RadICS, produced by RPC Radiy.

Technology FPGA (Field Programmable Gates Arrays) is one of the most intensively developed and applied in I&Cs of nuclear power plants (NPPs) and other safety & mission critical domains [23].

The FPGA-based chassis is a basic component of RTS architecture in the scope of this piece. Each chassis can contain up to 7 module types: analog and digital input modules (AIM, DIM); analog and digital output modules (AOM, DOM); logic module (LM); optical communication module (OCM); and analog input for neutron flux measurement module (AIFM). All modules are based on FPGA chips. The modules can be placed in 16 different positions on the chassis (two reserved positions for LM), using LVDS and fiber optical lines for internal/external communications. Such flexible redundancy management helps to ensure the high availability of the system. In this paper, we consider the chassis consisting of five modules: LM, DIM, DOM, AIM and AOM.

In section 4.1 we briefly describe the studied RTS and it reliability-block diagram, as the more detailed description is given in [24]. Section 4.2 presents the Markov chain for RTS and application of the metric-based approach and tools selection criteria. In section 4.3 shows the results of dependability and safety parameters assessment using received recommendations and obtained results verification.

A. *Description of the System Under Study*

The Fig. 6 presents the structure diagram of a typical chassis. It is assumed that the corresponding components of all the chassis in the channels are identical, i.e. DIM on the 1st chassis is identical to the same module on other chassis in the channels, etc. The failure of the LM leads to the failure of the whole chassis, and failures of the DIM, DOM, AIM, AOM result in chassis malfunction. Therefore, it was assumed that failure of any module implies the general failed state of the chassis.

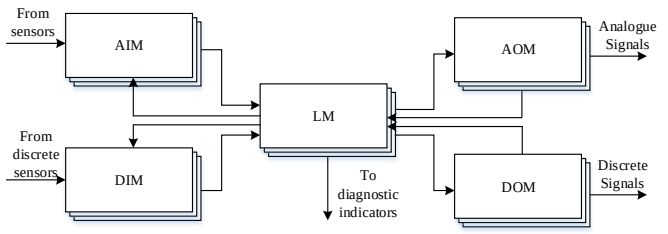

Fig.6. The strucure diagram of a typical chassis

Fig. 7 presents the RBD for the three-channel two- chassis architecture.

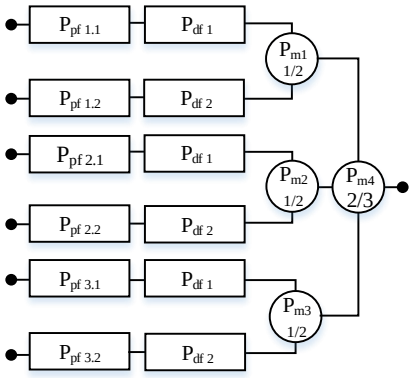

Fig.7. Reliability-block diagram of three-channel two-chassis RTS

All chassis in the channels have identical hardware structures, but the software run on the system channels is diverse [10], i.e. non-identical but functionally equivalent software copies are deployed on the system channels. Each channel independently receives information from sensors and other NPP systems. The channels, each being capable of forming a reactor trip signal, are independent.

Reliability index $P_{pfi.j}$ determines hardware reliability of the chassis $T_{i.j}$ (defined by physical faults), where i indicates main ($T_{1.j}$) or diverse ($T_{2.j}$) channels, and j indicates the chassis number. Reliability index P_{dfi} determines software reliability of the main or diverse channels (defined by software faults), where i indicates the channel. Reliability index P_{mi}, determines reliability of the majority element m_i, where $i \in \overline{(1,4)}$.

B. *Markov chain for the RTS*

Let us further denote by λd and λp the design and physical failure rates, respectively, and by μd and μp the repair rates after design and physical failures. The MC for RTS system is shown on Fig. 8.

The detailed model construction and description is shown in [24].

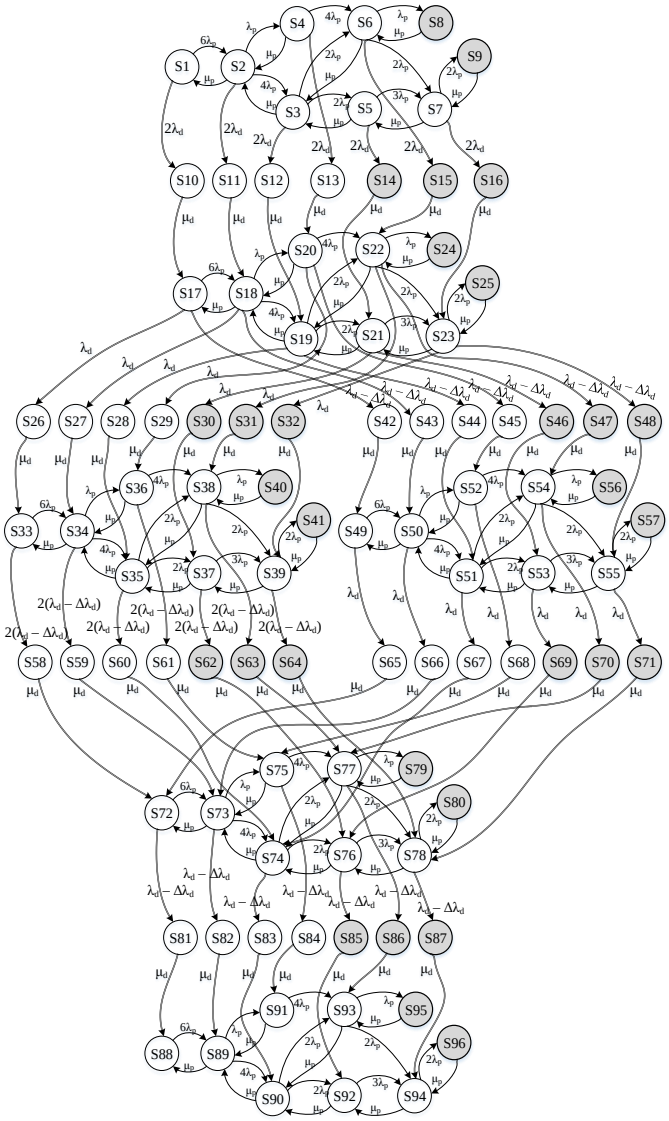

Fig.8. The multi-fragmental model for three-channel two-chassis RTS architecture

We use the following assumptions to create the MC for observed RTS:

- Each element of the research system in random moment of time can be only in two states – working and failure.
- The systems control and majority elements provide unstoppable correct functioning.
- The system maintenance is performed by one group of engineers, thus failed chassis are repaired sequentially. It should be noted, that recovering strategy use the maximal number of working channels in a priority.

- All detected defects are eliminated instantaneously and no new defects are introduced. The mean time between failures and mean time to repair are exponentially distributed [13].
- Software testing datasets are updated after each test. The testing is performed on the complete body of input data.
- The observed RTS is FPGA-based, thus investigated software faults are such kinds of faults, which are typical for VHDL coding process that were not covered by V&V procedure. The architecture-level MC shows the rare kind of design faults that can cause a general system failure, thus we expect that not more than two undetected design faults on each software version [13, 25].
- The failure rate of the design faults $\lambda_{d(i)}$ is proportional to their residual amount n_i in i – different software versions [13]. This assumption uses an incremental change of the software failure rate after detected design fault elimination ($\lambda_{d(i)}$ vary on a constant $\Delta\lambda_{d(i)}$). Such failure rates can be presented using multi-fragmentation approach [22].
- The design failures on diverse software versions are independent events, but equal in severity. Thus, we assume that failure and repair rates for the failures caused by design faults are equal.

$$\lambda_{d1}=\lambda_{d2}\Rightarrow\lambda_d=\lambda_{d1}+\lambda_{d2},\quad \mu_{d1}=\mu_{d2};\mu_d=\lambda_d/(\sum_{i=1}^{2}\frac{\lambda_{d(i)}}{\mu_{d(i)}}) \tag{12}$$

The neglecting of this assumption will increase the resulting MC state-space, but still we can apply the MFM principle for it construction. Thereby, the assumption was used in a purpose of reducing the model size. To check the developed model sensitivity we use four sets of parameters values, which are presented in Table 2. The results of testing the stiffness (1), decomposability (4), sparsity (5) and fragmentedness (6) characteristics are as follows.

- Stiffness: moderately stiff with *s(x) = 0.167*.
- Decomposability: completely decomposable (CD) with *E = 0.02*.
- Sparsity: highly sparse with *ms = 0.2*.
- Fragmentedness: moderately fragmented with N_{fr}*=6*.

TABLE II. MARKOV'S CHAINS PERAMETER VALUES

	λ_d (1/h)	$\Delta\lambda_d$ (1/h)	λ_p (1/h)	μ_d (1/h)	μ_p (1/h)	t (h)
1	10^{-5}	$5\cdot10^{-6}$	10^{-4}	0.01	1	[0; 30 000]
2	$2.5\cdot10^{-5}$	$1.25\cdot10^{-5}$				
3	$5\cdot10^{-5}$	$2.5\cdot10^{-5}$				
4	$7.5\cdot10^{-5}$	$3.75\cdot10^{-5}$				

With the metric-based diagram (Fig. 5), we receive the recommendation to use tolerance approach as the main solution technique and verify obtained result with avoidance approach. As the model appears to be moderately stiff, we need to use stiffness-stable numerical methods (STA).

We have consequently apply tools selecting criteria 1 – 3 over the M = {Matlab, Mathematica, Maple, SHARP, ToolKit Markov} under the following preferences:

- manual (graphical or matrix) MC construction;
- export (import) of underlying matrix of Kolmogorov differential equations to (from) the text editor;
- functions for stiffness, sparsity and decomposability analysis;
- implemented numerical methods for transition measures calculation with $\varepsilon_1 \geq 10^{-6}$, $C_{max} = 3$;
- C_b equal to the prices of academic license of the corresponding software packages.

As the result we have obtained recommendation for Mathematica and/or Matlab application.

C. Solution and Result Verification

We use the recommended approaches to assess the RTS unavailability function, which also defines the *PFH* measure [2]. The unavailability function *U(t)* is defined as a sum of failed states probabilities, with initial condition *U(0) = 0*:

$$U(t)=1-A(t)=\sum_{i=1}^{n}P_i(t), i\in N, \tag{13}$$

where, *A(t)* is RTS availability function.

Due to recommendations we use the STA as a main techniques, thus the *U(t)* for four sets of RTS parameter (Table 1) was calculated using build-in function of implicit RK in Mathematica. The results are shown on Fig. 9.

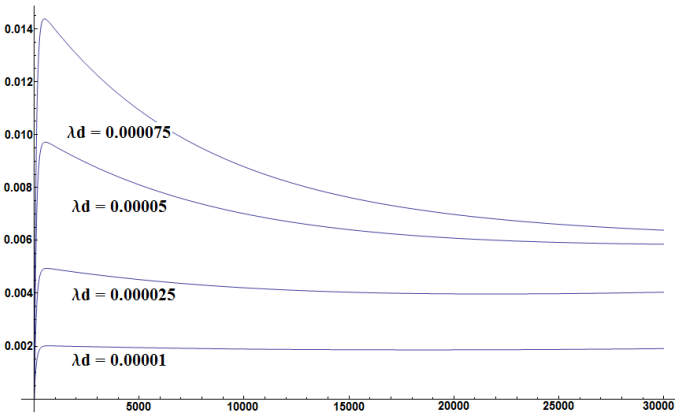

Fig.9. U(t) calculated using STA in Matheamtica

The *U(t)* result were verified using Matlab inner function, that also implements the implicit RK algorithm and SAA, particularly aggregation/ disaggregation technique [11].

The result of *U(t)* verification for $\lambda_d = 5 \cdot 10^{-5}$ is presented on Fig. 10.

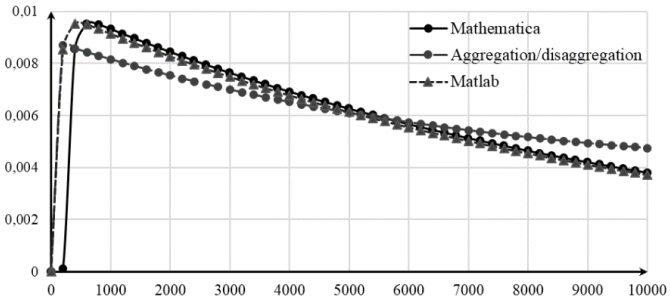

Fig.10. U(t) results verification

V. CONCLUSION

This paper presents the metric-based approach for selection of the applicable solution approach, based on the analysis of basic MCs characteristics (stiffness, largeness (decomposability, irreducibility), sparsity, fragmentedness) and three criteria of tools selection aimed to take into account different types of implemented functions, accuracy of embedded numerical methods and overall price of the tool.

The paper describes a case study for assessment the unavailability parameter of NPP I&C system, in particular, RTS. We present the RBD and MC system models and use the multi-fragmentation principle to describe the complex hardware-software interconnection on the architecture level. The metric-based approach and software packages selection criteria were applied to make an informed selection of the solution technique and tool. We have tested the RTS unavailability function for different parameter sets (Table 2), using the main recommended approach – STA. The results were also verified by SAA. Based on the obtained U(t) values we can conclude that under all assumption presented in Section 4, the studied architecture of RTS constructed on the FPGA-based digital platform, provides the needed dependability and safety level.

In our future work we intend to calculate the risk function for presented RTS architecture, by eliminating the assumption that design failures are equal in severity and develop a tool to support the MC-based safety assessment using suggested metric approach and procedure of technique and tool selection, that minimize the risks of inaccurate calculations.

REFERENCES

[1] IAEA, 1999, “Modern Instrumentation and control for NPP: A guidebook”, No. 387.

[2] IEC 61508, 2010, “Functional Safety of Electrical/Electronic/Programmable Electronic Safety- Related Systems”.

[3] P. Buchholz, et al., 2004, “Approximate Computation of Transient Results for Large MC”, In proc. of IEEE QUEST’04, pp. 126-135, 2004.

[4] W. E. Smith, et al., “Availability Analysis of Blade Server Systems”, IBM Systems Journal, Vol. 47(4), pp. 1- 20, 2008.

[5] IEC 61165, 2008, “Application of Markov techniques”.

[6] V. Kharchenko, et. al.,“Markov’s Model and Tool-Based Assessment of Safety-Critical I&C Systems: Gaps of the IEC 61508”, In Proc. 12th Int. Conf. PSAM, p. 16, 2014.

[7] K. S. Trivedi, et al., “Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains”, Microelectronic Reliability, vol. 34(11), pp. 1825-1841, 1994.

[8] A. Reibman, A., “Numerical Transient Analysis of Markov models”, Comput. Opns. Res., Vol.15(1), pp. 19-36, 1988.

[9] W. S. Barge, et al., “Autonomous Solution Methods for Large Markov Chains”, Pennsylvania State University CiteSeerX Archives, p. 17, 2002.

[10] V. Kharchenko, et. al., “Availability Assessment of Computer Systems Described by Stiff Markov Chains: Case Study”, Springer, CCIS(412), pp. 112 – 135, 2013.

[11] A. Bobbio, et al., “A Aggregation Technique for Transient Analysis of Stiff Markov Chains”, IEEE Transactions on Computers, C-35, pp. 803-814, 1986.

[12] O. Arushanyan, et. al., “Numerical Solution of Ordinary Differential Equations using FORTRAN”, Moscow State University, Moscow, p. 336, 1990.

[13] V. Butenko, “Modeling of a Reactor Trip System Using Markov Chains: Case Study”, Proc. of 22nd ICONE, vol. 5, 2014.

[14] W. H. Sanders, et al., “Optimal State-space Lumping in Markov Chains”, Inf. Process. Lett., vol. 87(6), pp. 309-315, 2003.

[15] A. Srinivasan, et al., “Algorithms for Discrete Functions Manipulation”, In Proc. Int’l Conf. on CAD (ICCAD’90), pp. 92-95, 1990.

[16] R. E. Bryant, R. E, “Graph-based Algorithms for Boolean Function Manipulation”, IEEE Trans. Comp., vol. 35(8), pp. 677–691, 1986.

[17] H. R. Gail, et al., “Calculating Availability and Performability Measures of Repairable Computer Systems”, Journal of the ACM, vol. 36, pp. 171–193, 1989.

[18] W. H. Sanders, et al., “An Efficient Disk-based Tool for Solving Large Markov Models”, Performance Evaluation, vol. 33, pp. 67–84, 1998.

[19] P. J. Courtois, “Decomposability: Queueing and Computer Applications”, Academic Press, New York, p. 201, 1977.

[20] T. Cormen, “Introduction to Algorithms”, MIT Press, Computers, p. 180, 2001.

[21] W. H. Press, et al., “Numerical Recipes. The Art of Scientific Computing, 3rd Edition”, Cambridge University Press, p. 1260, 2007.

[22] V. Kharchenko, et al., “Multi-fragmental Availability Models of Critical Infrastructures with Variable Parameters of System Dependability”, Int. Journal Information & Security, vol 28. pp. 248 – 265, 2011.

[23] V. Kharchenko, A. Siora, V. Sklyar et al, “Multi-Diversity Versus Common Cause Failures “FPGA-Based Multi-Version NPP I&C Systems”, Proc. of the 76th Conf. NPIC&HMIT, Las-Vegas, Nevada, USA, 2010.

[24] V. Kharchenko, et. al., “Assessment of the Reactor Trip System Dependability: Two Markov’s Chains – Based Cases”, Proc. Of the 10th Int. Conf. DT, Zilina, Slovakia, pp. 103 – 109, 2014.

[25] W. Ehrlich, et al., "Applying Reliability Measurement: A Case Study". IEEE Software, March 1990, pp. 56-64, 1990.

[26] Miettinen, K., et. al. “Multiobjective Optimization. Interactiva and Evolutionary Approaches”, Springer-Verlag, Berlin, Heidelberg, p. 461, 2008.

[27] V. Kharchenko et.al, “Metric-based approach and Tool for Modeling the NPP I&C System Using Markov Chains”, Proc. of 23nd ICONE, accepted for publishing, 2014.