Markov's Model and Tool-Based Assessment of Safety-Critical I&C Systems: Gaps of the IEC 61508

V. Butenko[a], V. Kharchenko[a,b], O. Odarushchenko[b], P. Popov[c], V. Sklyar[b] and E. Odarushchenko[d]

[a] National Aerospace University "KhAI", Kharkiv, Ukraine
[b] Research and Production Company "Radiy", Kirovograd, Ukraine
[c] Centre of Software Reliability, City University London, London, United Kingdom
[d] Poltava National Technical University, Poltava, Ukraine

Abstract: The accurate dependability and safety assessment of systems for critical applications is an important task in development and certification processes. It can be conducted through probabilistic model-based evaluation using the variety of tools and techniques (T&T). As each T&T is bounded by its application area the careful selection of the appropriate one is highly important. In this paper, we present the gap-analysis of well-known modeling approach – Markov modeling, mainly for T&T selection and application procedures, and how one of the leading safety standard IEC 61508 tracks those gaps. We discuss how main assessment risks can be eliminated or minimized using metric-based approach and present the safety assessment of typical NPP I&C system, the Reactor Trip System. The results analysis determines the feasibility of introducing new regulatory requirements for selection and application of T&T, which are used for MM-based assessment of safety.

Keywords: Markov chains, standard, metric, stiffness, largeness, sparsity.

1. INTRODUCTION

Dependability and safety assessment of systems for critical applications, such as NPP I&Cs, is an essential part of the development and certification processes as it either allows for demonstrating that relevant regulations have been met and for making informed decisions about the risks and consequences of inaccurate assessment results.

Dependability and safety of such complex systems can be assessed through model-based evaluation supported by specialized tools (λPredict, Möbius, SHARP, etc.), off-the-shelf tools (Maple, Matlab, Mathematica, etc.) and own developed utilities (ODU) [9]. Such model-based evaluation can be performed through discrete-event simulation (DES), analytic models or combining simulation and analytical approaches. The main advantage of DES is ability to consider detailed system behavior in the model, while the drawback is long execution time, when accurate solution is needed. Analytical models tend to be more abstract, but are easier to develop and faster to solve than a DES models. The main difficulty is a necessity to set additional assumptions to make the models tractable [1]. Analytical modeling techniques can be split into two groups: state space (Markov chains, Petri nets, etc.) and non-state space (RBD, FTA, etc.) techniques. Selection of the appropriate technique is provided based on measurements of interest, level of components detalization, etc.

One of the leading standards in the safety area IEC 61508-2010 provides no special requirements for techniques and tools (T&T), which are used to evaluate the system safety indicators, excepting the strong recommendation, that practitioner must have an understanding of the techniques used by software package to ensure its use is suitable for the specific application. The absence of special requirements can be explained by long-time use of the mentioned techniques and proved by their detailed description in IEC 61508-2010 (part 6) [2]. In contrast to the T&T, many requirements were developed for I&Cs verification and validation (V&V) tools are compatible by strength to the requirements of produced software and systems (see standard IEC 60880-2006 [3]).

The state space models are always preferred to non-space models, if it is important to model such complex situations as failure/repair dependencies, shared repair facilities [1] or provide the detailed presentation of system behavior for communication with engineering teams [4]. One of the main computation difficulties, in particular for Markov chains, is the size of model's state space, which increases exponentially when the number of components of the system under study expands [2].

System modelers are often interested in transient measures, which provide more useful information than steady-state measures. Modeling components interaction and interdependencies expands the model significantly, thus making the precise computation of system transient measures almost infeasible. Whilst numerical methods and imitation modeling can be applied to handling this problem, they are also limited by model size and such difficulties as stiffness [5] and sparsity [6]. Stiffness is a well-known undesirable property of many practical MCs [7] as it poses a problem of finding transient solutions. Stiffness in models is caused by: i) in case of repairable systems the rates of failure and repair differ by several orders of magnitude [7]; ii) fault-tolerant computer systems (CS) use redundancy. The rates of simultaneous failure of redundant components are typically significantly lower than the rates of the individual components [7]; iii) in models of reliability of modular software the modules' failure rates are significantly lower than the rates of passing the control from a module to a module [7]. Sparsity [8] corresponds to systems, which are loosely coupled. In the subfield of numerical analysis, a sparse matrix is a matrix populated primarily with zeros [10]. If the MC is large it becomes wasteful to reserve storage for zero elements, thus solution methods that do not preserve sparsity, is unacceptable for most large problems [6].

Several approaches were developed to deal efficiently with MC largeness [12, 13] and stiffness [14, 5, 7]. In both cases, they can be split into two main groups – "avoidance" and "tolerance" approaches. The avoidance approach overcomes largeness by exploiting the certain properties of the model to reduce the size of underlying MC [12]. In the largeness tolerance approach the new algorithms are designed to manipulate large MC, and special data structures are used to reduce state transition matrix, iteration vector, etc [13]. For stiff models, with tolerance approach (STA) the *specialised numerical methods* [6, 14, 15] are used to provide highly accurate results despite stiffness. The limitations of STA are: i) STA cannot deal effectively with large models, and ii) computational efficiency is difficult to achieve when highly accurate solutions are sought. The stiffness avoidance (SAA) solution, on the other hand, is based on an *approximation algorithm* which converts a stiff MC to a non-stiff chain first, which typically has a significantly smaller state space [7]. An advantage of this approach is that it can deal effectively with large stiff MCs.

The variety of T&T [9] is extremely helpful in the process of system modeling but this also poses a difficulty when it comes to choosing the most appropriate method for a specific assessment. As every tool is limited in its properties and applicability, a careful selection is needed for the tools used to solve large and stiff MCs accurately and efficiently. It should be noted that stiffness usually requires the modeler to focus on a number of math details to avoid the use of inefficient approaches, methods [15], and tools. This view goes against the recommendations of IEC 61508-6 [2]. This standard asserts that methods for solving Markov models have been developed long ago and trying to improve these methods does not seem sensible. The previous works show [9] that solving a large and/or stiff Markov model requires a careful selection of the solution method/tool. Otherwise, the results can differ in several orders of magnitude [9], thus, use of inappropriate method/tool for the solution of a non-trivial MC may lead to significant errors.

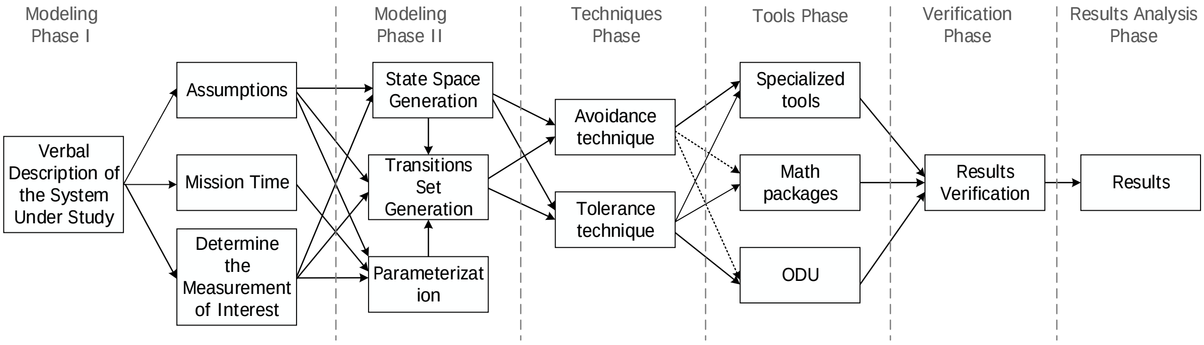
In this paper we present the detailed analysis of the main gaps in Markov modeling approach, mainly in T&T selection and application, and how they are traced by few well-known standards in the safety area (IEC 61508:2010 [2], IEC 60300:2003 [16], IEC 61165:2006 [17] and ECSS-Q-ST-30-09:2008 [4]). By the "*gap*" we define special risks, which are accompanies the MC application procedure.

We discuss the ways how those risks can be eliminated or minimized and present the case study for safety assessment of a typical NPP instrumentation and control system (I&Cs), the Reactor Trip System (RTS) based on a digital platform produced by RPC Radiy. This is a two-channel FPGA-based system with three parallel tracks (sub-channels) of voting logic “2-out-of-3” in each of the channels. The rest of the paper is organized as follow: in section 2 we presents the analysis of the main gaps (i.e. risks) of MC assessment and how they are tracked by standards; in section 3 we describe the metric-based approach, which can be used to decrease the risks and to ensure the results accuracy; in section 4 we presents the industrial case study using the metric-based approach, to demonstrate its usefulness; in section 5 we present the conclusions and outline the problems left for future consideration.

2. GAP-ANALYSIS OF MC-BASED ASSESSMENT

In this section, we provide the detailed analysis of the main *gaps* (i.e. risks) of MM approach and how they are tracked by the standards. Overall, application of MM approach can be presented as a sequence of six stages (Fig. 1), namely: first and second modeling phases, techniques phase, tools phase, verification and results analysis phases.

Fig. 1. Sequence of stages in MM approach



The *first modeling phase* presents the processes of verbal system description, definition the measures of interest, development of modeling assumptions and determination of the studied time interval. The most typical gaps of this stage and how they are reviewed by standards are presented in Table 1.

Table 1: Gaps of the first modeling phase

Gap	Reviewed by Standard
Confirm the applicability of the Markov modeling approach for the specific case	**IEC 60300-3-1** “…selection of method for dependability analysis is individual and performed by joint effort of dependability experts and system experts. The selection has to be performed on the early stages of system development and analyzed on applicability… ”
Introduce the assumptions, which will keep the required abstraction level with respect to the calculated parameters	The main assumptions used for MC approach are presented in IEC 61165 (section 6) and IEC 61508 – 6 (section B.5.2).

During the *second modeling phase*, researcher determines the model parameters, using statistic data, etc. and performs generation of system state space and set of transitions taking into account designed modeling assumptions. The main gaps of this stage and their tracking by standards are listed in Table 2. It should be noted, that MC approach described in standards IEC 61508 (part 6) [2] and IEC 61165 [17] assumes constant time-independent state transition rates, while in practice those rates can vary over time due to physical aging errors, etc.

Table 2: Gaps of the second modeling phase

Gap	Reviewed by Standard
The MM approach requires construction of the full set of all possible system states, which exponentially increase the model	**IEC 61508 – 6** “…The main problem with Markov graphs is that the number of states increases exponentially when the number of

size.	components of the system under study increases…” **ECSS-Q-ST-30-09** “…However, the system complexity can generate a high number of expected states that have impact on the calculation aspects (time and accuracy)...” **IEC 60300-3-1** “…accounting the additional system components exponentially increases the state space and complicates analysis… ”
Analyze the statistic data to determine system parameters	The detailed explanation is presented in IEC 61508 (6 and 7 parts)

On the *techniques phase*, using the analysis of constructed state space the researcher choose the assessment technique. If the model appears to be large and/or stiff and/or sparse, a careful selection of the solution approach is required. Using an inappropriate method for the solution of a non-trivial MC may lead to significant errors [9]. The main gaps of this phase are shown in the Table 3. We also noted the following ambiguity between IEC 61508 and IEC 61165: while the IEC 61508 asserts that modeler should not focus on the underlying mathematical details, the IEC 61165 requires the help of experts in applied mathematics area during model solution.

Table 3: Gaps of the techniques phase

Gap	Reviewed by Standard
Choose the appropriate solution approach/method based on its applicability for the specific case, accuracy and level of confidence.	**IEC 61508 – 7** “… a homogeneous Markov graph is only a simple and common set of linear differential equations with constant coefficients. This has been analyzed for a long time and powerful algorithms have been developed and are available to handle them…”
In case of transient solution, make the analysis of such mathematical details as stiffness, sparsity etc., which can influence on the achieved results.	**IEC 61508 – 6** “…Efficient algorithms have been developed and implemented in software packages a long time ago in order to solve above equations. Then, when using this approach, the analyst can focus only on the building of the models and not on the underlying mathematics…” **IEC 61165** “..the solution methods can be quiet complex, thus the specialized software packages and/or experts in applied math area are required…”

During the *tools phase* researcher selects appropriate software package (tool) to obtain the MC solution using the chosen technique. The Table 4 presents the list of main gaps of current phase and how they are reviewed by standards. It is important to note, that the human-based errors can be introduced during the MC manual construction using the means of selected tool. In addition, tool must support portability of a solution project, so researcher can use it in additional calculations, thus the usability-oriented selection of tools is needed.

Table 5: Gaps of the tools phase

Gaps	Reviewed by Standard
Using the selected solution approach/ method choose the efficient and highly trusted software package.	**IEC 61508 – 6** “…If software programs are used to perform the calculations then the practitioner shall have an understanding of the formulae/techniques used by the software package to ensure its use is suitable for the specific application…” **IEC 60300-3-1** “…m) tools performance. Are the tools user friendly? Do they share the interface with other tools, so the results can be transmitted for multiple use?...”
Construct/generate and solve the MM. For	**IEC 61508 – 6**

the model manual construction, it is important to verify the resulting MC to detect the human-based errors.	"…When dependencies between components cannot be neglected, some tools are available to build automatically the Markov graphs. They are based on models of a higher level than Markov models (e.g. Petri nets, formal language)…"

On the *verification phase* results are checked to ensure confidence and accuracy of the obtained values. We note that standards mainly recommend the manual results verification. In most cases the size of state space makes the manual calculations infeasible in most cases, thus verification can be supported by another tools or approaches.

Table 6: Gaps of the verification phase

Problem	Covered by Standard
Verify the results using manual calculations or another SW package.	**IEC 61508 – 6** "…The practitioner should also verify the software package by checking its output with some manual calculated test cases…" **IEC 61508 – 7** "…when anything but the simplest calculations is performed in floating point, the validity of the calculations must be checked to ensure that the accuracy required be the application is actually achieved…" **IEC 60300-3-1** "…l) check the trustworthiness. Can we check results manually? If not, the software package is user friendly?..."

The gaps of techniques, tools and verification phases have significant impact on the results accuracy, and observed standards does not exhaustively track all risks. Thus, it is important to know the ways in which the risk of calculating the incorrect resulting value by using inappropriate T&T can be decreased or eliminated.

3. METRIC- BASED APPROACH FOR DECREASING THE ASSESSMENT RISKS

In this section, we discuss how the gaps (risks) of solution technique selection (*technique phase*) can be minimized/eliminated using metric-based approach. We also present the brief overview of the "largeness-tolerance" (LTA), "largeness-avoidance" (LAA), "stiffness-tolerance" (STA) and "stiffness-avoidance" (SAA) approaches.

The model largeness, stiffness and sparsity can complicate the solution process and use of inefficient method can lead to the inaccurate results [9]. There are at least four important considerations for making informed decision between different solution techniques: efficiency and applicability of an algorithm, the structure of a matrix, size and storage needs and the architecture of the computer used for MC solution [18]. The largeness property force to use additional storage place, stiffness and largeness properties can influence of the efficiency and applicability of a numerical solution algorithm and sparsity affects the structure of a matrix. Analyzing MC on the presence of these properties (further metrics) can help in choosing the effective and convenient for transient solution T&T, thus decrease the assessment risks. Work [18] presents selection approach for finding the steady-state solutions of large MC.

3.1. Largeness

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 LAA approach the certain properties of model representation are exploited

to reduce the size of the MC to obtain the measures of interest [12]. The state-level and model-level [12] 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 LTA are designed to manipulate large MC using special algorithms and data structures to reduce and store transition probabilities matrix [13]. The numerous works [13, 19] 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 [20, 21]. Analysis of MC irreducibility and decomposability properties can help to make a prior selection between described techniques [18].

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 (1), which has a matrix with non-zero elements in off-diagonal blocks are small compared with those in the diagonal blocks [22]:

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

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 is of form (2), where E contains all of-diagonal blocks. The quantity (3) is referred to as degree of decomposability [22]. If $E = 0$ then MC is said to be completely decomposable (CD).

$$A = diag(A_{11}, A_{22,}, \dots, A_{nn}) + E \qquad (2)$$

$$\|E\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} | e_{ij} | \qquad (3)$$

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 []. Determination of the strongly connected components can also help to provide accurate model reduction.

3.2. Stiffness

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 (4):

$$s(x) = \frac{\max\limits_{i=1,n} | \mathrm{Re}(\lambda_i) |}{\min\limits_{i=1,n} | \mathrm{Re}(\lambda_i) |} >> 1 \qquad (4)$$

where $s(x)$ denotes the stiffness index and λ_i are the eigenvalues of the Jacobian matrix ($\mathrm{Re}\,\lambda_i < 0, i = 1,2,\dots,n$)[14]. The previous empirical work [9] 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. The $s(x)$ values can be split into three groups: high $s(x) \ge 10^4$, moderate $10^2 < s(x) < 10^4$ and low $s(x) < 10^2$ [9]. The basic methods to overcome stiffness are described next.

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 [7], [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 [7]. 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.

Stiffness-tolerance approach. The main idea of this approach is using methods that are stable for solving stiff models. These 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.

(a) The classical (non-modified) numerical methods for solving stiff DEs use special single-step and multi-step integration methods. Examples of such methods are the implicit Runge-Kutta, the TR-BDF2, the Rosenbrock method, the exponential method, the implicit Gir method described in [14], [15], [6]. The implicit Runge-Kutta, TR-BDF2 and Rosenbrock method are implemented by several mathematical off-the-shelf software packages and are usually considered the most accurate methods for solving stiff ODEs.

(b) An example of the modified numerical methods is the exponential modified method. The original algorithm was presented in [9,14] and is based on the evaluation of the matrix exponent. In [14] this method is recommended as one of the most effective algorithms for solving the class of ODE systems with a high value of the Lipchitz constant, and as a special part of a stiff ODE. As a modification part, an automated adaptive step of integration can be implemented. As the method has a multi-step algorithm the given modification can increase the accuracy of the solution [9, 14].

The solution provided by using any numerical method is expected to be accurate. However, typically the result obtained using numerical method include errors coming from different sources, such as *truncation error*, *round-off error, initial error* etc.

3.3. 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 [6]. 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 [23]. Paper [18] 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 (5):

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

Table 7 presents the recommendation for prior selection of the transient solution approach based on analysis of such MC metrics as stiffness and largeness, in particular stiffness index, size of MC state space and decomposability property. If both approaches can be used, the symbol "✓✓" stresses the probably more preferable one. Detection of the strongly connected graph components (irreducibility property) can help in determining sub-sets for approximate aggregation technique. In [18] authors studied the influence of *ms* value on accuracy of the direct techniques for steady-state MC solution, and recommended to give additional attention on *fill-in* amount for matrices with $n \geq 500$ and $ms > 0.85$. The *fill-in* is a property when initially zero matrix elements become nonzero during solution process and for which storage must be reserved. Thus for large MC evaluation of *ms* value helps to decide whether the special sparse structures are required or not.

Table 7: Prior selection of the solution approach for MC transient analysis

Metric			Avoidance approach	Tolerance approach
Large MC	High stiff	CD or NCD	✓	
		Non decomposable		✓
	Moderate stiff	CD or NCD	✓	
		Non Decomposable		✓
	Low stiff	CD or NCD	✓✓	✓
		Non Decomposable		✓
No-large MC	High stiff	CD or NCD	✓✓	✓
		Non Decomposable		✓
	Moderate stiff	CD or NCD	✓	✓✓
		Non Decomposable		✓
	Low stiff	CD or NCD	✓	✓✓
		Non Decomposable		✓

4. CASE STUDY: NPP I&C SYSTEM

4.1. Research system description

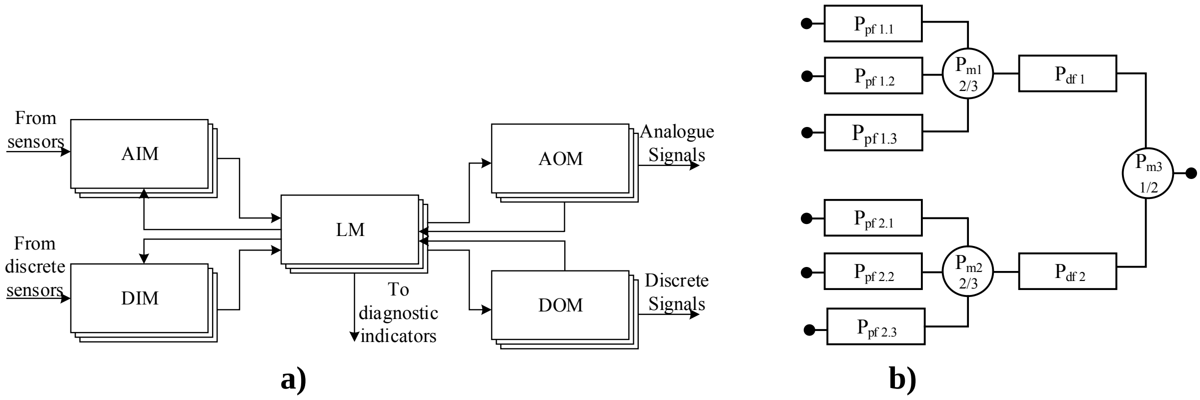
This section presents the case study for dependability assessment of typical NPP I&C system produced by RPC Radiy. This is Reactor Trip System (RTS) with two-channel, three-track architecture, on voting logic “2-out-of-3” for tracks in each channel and “1-out-of-2” between channels. The FPGA-based track is a basic component of observed RTS. Generally, each track can contain up to 7 module types: analogue and digital input modules (AIM, DIM); analogue and digital output modules (AOM, DOM); logic module (LM); optical communication module (OCM); and analog input for neutron flux measurement module (AIFM). The modules can be placed in 16 different positions on the track (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. Each channel independently receives information from sensors and other NPP systems. The channels, each being capable of forming a reactor trip signal, are independent.

In this paper, we consider the tracks consisting of five modules: LM, DIM, DOM, AIM and AOM. The Fig. 2.a presents the structure diagram of a typical track. It is assumed that the corresponding components of all the tracks in the channels are identical, i.e. DIM on the 1st track is identical to the same module on other tracks in the channels, etc. The failure of the LM leads to the failure of the whole track, and failures of the DIM, DOM, AIM, AOM result in track malfunction. Therefore, it was assumed that failure of any module implies the general failed state of the track.

The RDB for RTS is presented on Fig. 2.b. All tracks in the channels have identical hardware structures, but the software run on the system channels is diverse [24], i.e. non-identical but functionally equivalent software copies are deployed on the system channels.

Reliability index $P_{pfi.j}$ determines hardware reliability of the track $T_{i.j}$ (defined by physical faults), where i indicates main ($T_{1.j}$) or diverse ($T_{2.j}$) channels, and j indicates the track 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 $i \in \overline{(1,3)}$ ·

Fig. 2. a) The structure diagram of a typical track; b) RBD of two-channel tree-track RTS



4.2. RTS Markov model. Physical and Design Faults

The following assumptions were used during the MM development.

- Each element of the research system at an arbitrary moment of time can only be in one of the two states – “working” and “failure”.
- The systems majority and control elements provide non-stop correct functioning.
- The failure rate of the design faults $\lambda_{d(i)}$ is proportional to their residual amount n_i in i – different software versions [9]. This assumption is based on the model[25] and allows evaluating an incremental change of the software failure rate after detected design fault elimination ($\lambda_{d(i)}$ vary on a constant $\Delta\lambda_{d(i)}$). It captures a plausible phenomenon, which is well accepted in practice: various software ‘aging effects’ are indeed modelled by changing rate of software failure.
- All detected faults are eliminated instantaneously and no new defects are introduced. The mean time between failures and mean time to repair are exponentially distributed. Not more than two undetected software design faults are expected in the resulting software.
- Software test sets of data are updated after each test. The testing is performed on the complete amount of input data.
- The priority recovering strategy is repairing back to two working channels.

In MMs the failure rate variation can be illustrated using *multi-fragmentation approach* [9]. Using this approach the model is presented as N fragments that have the same structure but may differ in one or more parameter values [9]. The number of fragments F in the MC depends on the number of expected undetected software faults n_i in i – different software versions (6).

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

The model parameters are as follows:

- $\lambda_{p(i,j)}$, $\mu_{p(i,j)}$, where $i \leq 2$ and $j \leq 3$ - the failure and repair rates for the failures caused by physical faults in the track $T_{i,j}$. As each track consists of five module types, the $\lambda_{p(i,j)}$ and $\mu_{p(i,j)}$ of the track $T_{i,j}$ can be calculated using (7) and (8) respectively:

$$\lambda_p(i,j) = \lambda_{DIM(i,j)} + \lambda_{DOM(i,j)} + \lambda_{LM(i,j)} + \lambda_{AIM(i,j)} + \lambda_{AOM(i,j)} \tag{7}$$

$$\mu_p(i,j) = \lambda_p(i,j) / (\lambda_{DIM(i,j)} / \mu_{DIM(i,j)} + \lambda_{DOM(i,j)} / \mu_{DOM(i,j)} + \lambda_{LM(i,j)} / \mu_{LM(i,j)} + \\ + \lambda_{AIM(i,j)} / \mu_{AIM(i,j)} + \lambda_{AOM(i,j)} / \mu_{AOM(i,j)}), \tag{8}$$

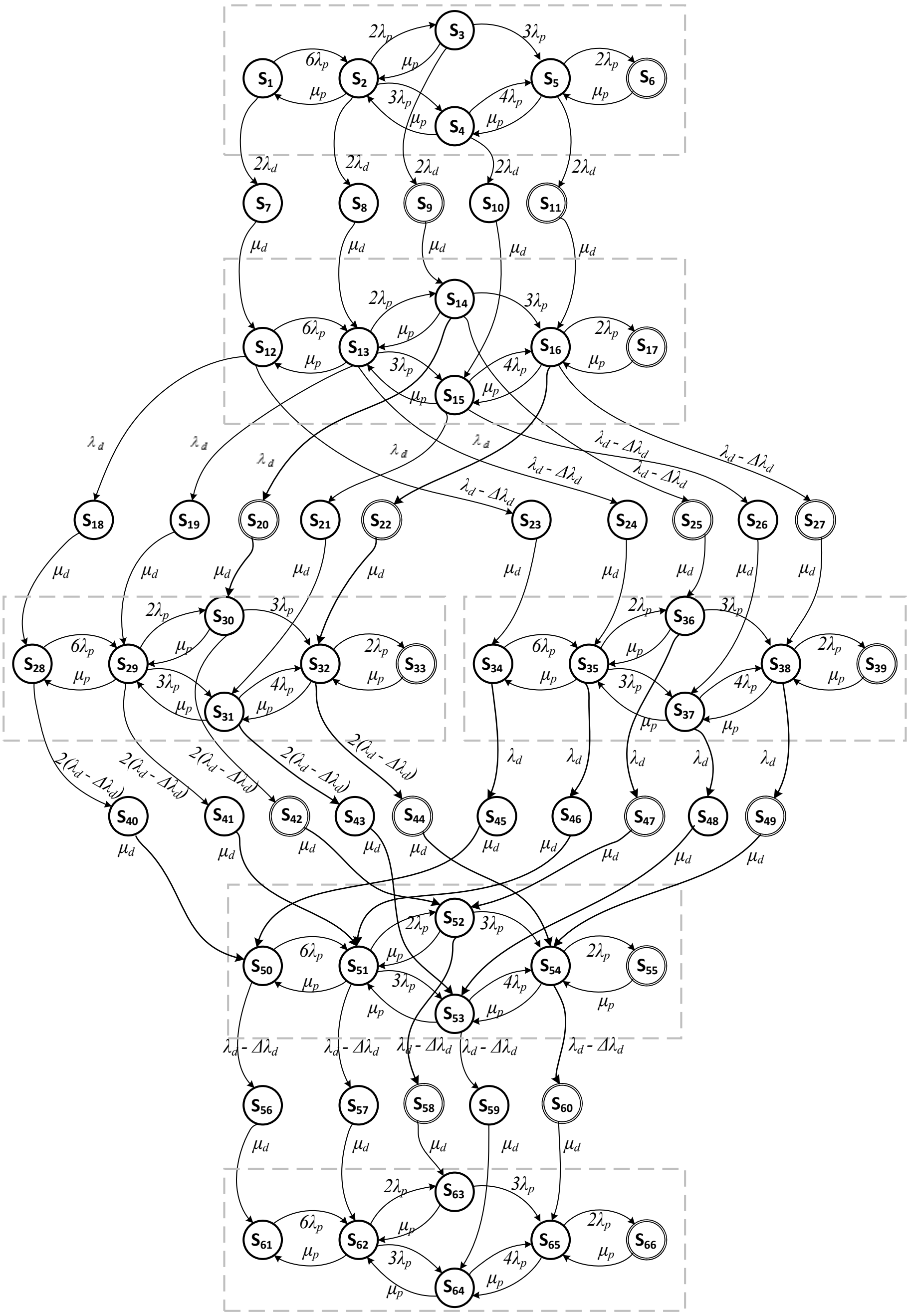
where $\{\lambda_{DIM(i,j)}, \lambda_{DOM(i,j)}, \lambda_{LM(i,j)}, \lambda_{AIM(i,j)}, \lambda_{AOM(i,j)}\}$ and $\{\mu_{DIM(i,j)}, \mu_{DOM(i,j)}, \mu_{LM(i,j)}, \mu_{AIM(i,j)}, \mu_{AOM(i,j)}\}$ are failure and repair rates caused by physical faults of DIM, DOM, LM, AIM, AOM, respectively. As all corresponding components of the tracks are identical, their failure and repair rates for the failures caused by physical faults are also equal. Thus, values of $\lambda_{p(i,j)}$, $\mu_{p(i,j)}$ are equal for all $T_{i,j}$ tracks.

- The failure and repair rates (9) for the failures caused by software design faults are equal. The steps of the failure rate decrease (after the channel recovery) are equal for both channels $\Delta\lambda_{d1} = \Delta\lambda_{d2}$ [9]:

$$\lambda_{d1} = \lambda_{d2} \Rightarrow \lambda_d = \lambda_{d1} + \lambda_{d2} \ \ and \ \ \mu_{d1} = \mu_{d2}; \mu_d = \lambda_d \, / (\sum_{i=1}^{2} \frac{\lambda_{d(i)}}{\mu_{d(i)}}) \qquad (9)$$

The MM for studied system taking into account physical and design faults is presented on Fig. 3.

Fig. 3. Multi-fragmental MM of RTS



4.3. Model Solution using Metric-based Approach

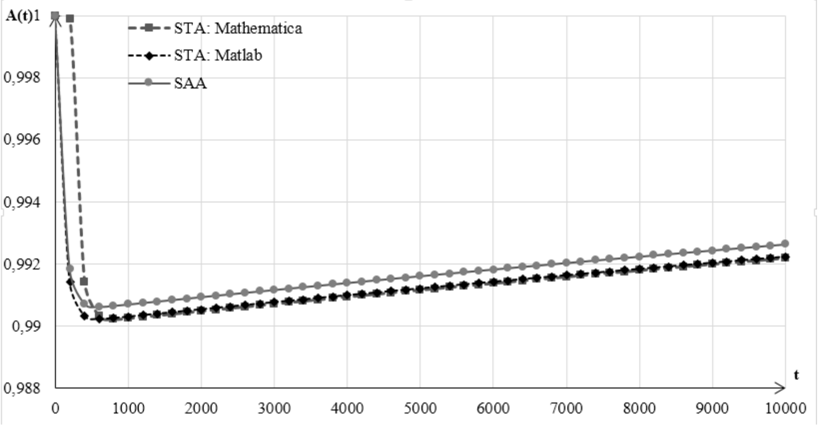
In this section we provide the solution of the MM (Fig.3) using the prior assessment of the MC properties – stiffness, decomposability and sparsity. The MM was solved for the values of $\lambda_p = 10^{-4}$, $\mu_p = 1$, $\lambda_d = 5\cdot10^{-5}$, $\mu_d = 0.01$, $\Delta\lambda_d = 2.5\cdot10^{-5}$ and an accuracy requirement of 10^{-6}. The MM consists of sixty-six states, thus can be referred to as no-large MC.

- *Stiffness index.* The MM is moderately-stiff with index (4): $s(x) = 1.667 \cdot 10^{-3}$ (8), as $max\ |Re(\lambda_i)| = 1.00063$ and $min\ |Re(\lambda_i)| = 0.0006$.
- *Decomposability*. The MM is NCD with **E** (3): $E = 0{,}02$.
- *Sparsity*. The matrix score was calculated using (5) *ms*: $ms = 0{,}13$. As MC is classified as no-large there is no need to use special sparse structures [18].

Using the prior selection procedure (Table 7) we can apply both solution approaches – avoidance and tolerance, but the tolerance approach is said to be preferable.

The assessment of availability function *A(t)* using STA was performed in mathematical packages Mathematica and Matlab using a built-in function implementing the implicit Runge-Kutta method. The solution was computed on the time interval of [0; 10 000] hours with a time step of $h=200$ hours. As the MM is stiff the SAA, in particular aggregation/disaggregation technique developed by K. S. Trivedi, A. Bobbio and A. Reibmann, was used to solve the model [7]. The comparison of the results obtained by STA and SAA approaches for both RTS architectures are presented on Fig. 8. The average differences |ω| between STA and SAA *A(t)* results are: Matlab –Mathematica – 0.0001035; Matlab – SAA – 0.0004001; Mathematica – SAA – 0.0004002.

Fig. 4. Comparison of STA and SAA approach results



6. CONCLUSIONS

There are some gaps in standards including IEC 61508 regarding techniques and tools choice and accuracy of indicators evaluation on MC-based assessment of safety-critical systems. We conclude that such difficulties as MC largeness, stiffness and sparsity affects significantly on the accuracy of the solution methods, and it is important to take into account such features while selecting the T&T. Analysis of the obtained results allows us to formulate a few problems regarding application of assessment tools: usability-oriented selection of tool in case of solving large MC; results accuracy and level of confidence; high quality verification of the obtained results.

These circumstances determine the feasibility of introducing the additional regulatory requirements to the choice of appropriate T&T and to verification of accuracy of the MC-based safety assessment using these T&T.

References

[1] S. Archana, R. Srinivasan, K. S. Trivedi. "*Availability Models in Practice*", Proc. Int. Workshop on Fault-Tolerant Control and Computing (FTCC-1), May 22-23, Seoul, Korea, 2000.

[2] Standard IEC 61508: 2010, "*Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-Related Systems*".

[3] Standard IEC 60800 Ed. 2.0: 2006, "*Nuclear power plants – Instrumentation and control system important for safety – Software aspects for computer-based systems performing category A functions*".

[4] Standard ECSS-Q-ST-30-09: 2008, "*European Cooperation for Space Standardization (ECSS): Availability analysis*".

[5] M. Malhotra, J. K. Muppala, K. S. Trivedi. "*Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains*", Microelectronic Reliability, Vol.34(11), 1994, pp.1825-1841.

[6] A. Reibman, K. S. Trivedi. "*Numerical Transient Analysis of Markov models*", Comput. Opns. Res., Vol.15(1), 1988, pp. 19-36.

[7] A. Bobbio, K. S. Trivedi. "*A Aggregation Technique for Transient Analysis of Stiff Markov Chains*", IEEE Trans. on Comp., C-35, 1986, pp. 803-814.

[8] G. H. Golub, C. F. Van Loan. "*Matrix Computations*", JHU Press, 1996, p. 694.

[9] V. Kharchenko, O. Odarushchenko, V. Odarushchenko, P. Popov, "*Availability Assessment of Computer Systems Described by Stiff Markov Chains: Case Study*", Springer, CCIS , Vol. 412, 2013, pp. 112 - 135.

[10] J. Stoer, R. Bulirsch. "*Introduction to Numerical Analysis*", Springer, 2002, p.732.

[11] A. Reibman, K. S. Trivedi, S. Kumar, G. Ciardo. "*Analysis of Stiff Markov Chains*", ORSA Journal on Computing, Vol.1(2), 1989, pp.126-133.

[12] S. Derisavi, H. Holger, W. H. Sanders. "*Optimal State-space Lumping in Markov Chains*", Inf. Process. Lett., Vol. 87(6), 2003, pp. 309-315.

[13] A. Srinivasan, T. Kam, S. Malik, and R.E. Brayton. "*Algorithms for Discrete Functions Manipulation*", In Proc. Int'l Conf. on CAD (ICCAD'90), 1990, p. 92-95.

[14] O. Arushanyan, S. Zaletkin. "*Numerical Solution of Ordinary Differential Equations Using FORTRAN*", Moscow State University, Moscow, 1990, p. 336.

[15] E. Hairer, G. Wanner, "*Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*"/ R. Bank, R. I. Graham, J. Stoer, R. Varga, H. Yserentant (editors), Springer, 2010, p. 631.

[16] Standard IEC 60300-3-3 Ed.2.0: 2008, "*Application guide – Analysis techniques for dependability – Guide on methodology*".

[17] Standard IEC 61165: 2006, "*Application of Markov technqiues*".

[18] W. S. Barge, W. J. Stewart. "*Autonous Solution Methods for Large Markov Chains*", Pennsylvania State University CiteSeerX Archives, 2002, p. 17.

[19] R. E. Bryant. "*Graph-based Algorithms for Boolean Function Manipulation*", IEEE Trans. Comp., Vol. 35(8), 1986, pp. 677–691.

[20] E. de Souza e Silva, H. R. Gail. "*Calculating Availability and Performability Measures of Repairable Computer Systems*", Journal of the ACM, Vol. 36, 1989, pp. 171–193.

[21] D. D. Deavours, W. H. Sanders. "*An Efficient Disk-based Tool for Solving Large Markov Models*", Performance Evaluation, Vol. 33, 1998, pp. 67–84.

[22] P. J. Courtois. "*Decomposability: Queueing and Computer Applications*", Academic Press, New York, 1977, p. 201

[23] W. H. Press, S. A. Teukolsky, W.T. Vetterling, B. P. Flannery B.P. "*Numerical Recipes. The Art of Scientific Computing, 3rd Edition*", Cambridge University Press, 2007, p. 1260.

[24] B. Littlewood, P. Popov, L. Strigini. "*Modelling Software Design Diversity - a Review*", ACM Computing Surveys, Vol. 33(1), 2001, pp.177 – 208.

[25] Z. Jelinski, P. L. Moranda, "*Software Reliability Research. Statistical Computer Performance Evaluation*"/ W. Freiberger (editor), Academic press, New York, 1972, pp. 365 – 484.