Chapter 7. Tool-oriented Availability and Safety Assessment of a Reactor Trip Systems using Multi-Fragmented Markov Modeling

Oleg Odarushchenko
Research and Production Corporation Radics, Ukraine
Vyacheslav Kharchenko
Centre for Safety Infrastructure-Oriented Research and Analysis, Ukraine
Valentina Butenko
National Aerospace University "KhAI", Ukraine
Elena Odarushchenko
Poltava State Agrarian Academy, Ukraine

ABSTRACT

Chapter 7 presents the gap-analysis of well-known modeling approach using Markov modeling (MM), mainly for technique and tool selection and application procedures, and how one of the leading safety standard IEC 61508 tracks those gaps. The task of accurate availability and safety assessment of a Reactor Trip Systems for Nuclear Power Plants Instrumentation and Control Systems (NPP I&C) applications is discussed in context of development and certification processes. Risks of inaccurate assessment can be eliminated or minimized using metric-based approach and present the safety assessment of typical NPP I&C system. The results analysis determines the feasibility of introducing new regulatory requirements for selection and application of techniques and tools, which are used for MM-based assessment of availability and safety.

Keywords: Markov modeling, stiffness, largeness, sparsity, fragmentedness, safety, availability, reactor trip system, metric-based analysis

INTRODUCTION

Availability 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 demonstrates that relevant regulations have been met and for making an informed decisions about the risks and consequences of inaccurate assessment results.

Such model-based assessment 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 DES models. The main difficulty is a necessity to set additional assumptions to make the models tractable (Trivedi, K.S. et al. 2000). Analytical modeling techniques can be split into two groups: state space (Markov models, 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.

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 (Trivedi, K.S. et al. 2000) or provide the detailed presentation of system behavior for communication with engineering teams Malhotra, M. et al (1994).

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. Use of Markov models (MM) for modeling the transient behavior of the complex system can lead to the number of description and computational difficulties.
One of the main computational difficulties is model *largeness* (i.e. structural complexity) (Buchholz, P. 1996), which leads to problems in its construction, storage and solution. Adding the software component behavior into the MM increases its size rapidly. Because of models size the closed-form solutions of transient measures become infeasible, in this case, modeler can rely on the numerical methods or imitation modeling. Modeling components interaction enlarge the state space significantly, and results in sparse matrices of differential equations (DE) coefficients.
Sparsity (Hurleyб N. et al. 2009) corresponds to systems, which are loosely coupled. In the subfield of numerical analysis, a sparse matrix is a matrix populated primarily with zeros (Press, W.H. et al. 2007). If the MM 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 (Press, W.H. et al. 2007).
The next complexity in solving large models that effect on numerical solution results is the model *stiffness* (Bobbio, A. et al. 1986). It is an undesirable property of many practical MMs as it poses difficulties in finding transient solutions. In practice, stiffness in models of complex computer systems is caused by (Bobbio, A. et al. 1986):

- in case of repairable systems the rates of failure and repair differ by several orders of magnitude;
- 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;
- 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.

Several approaches were developed to deal efficiently with MM largeness and stiffness (Malhotra, M. et al 1994, Bobbio, A. et al. 1986), Arushanyan, O. et al. 1990). 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 MM (Malhotra, M. et al 1994). In the largeness tolerance approach the new algorithms are designed to manipulate large MM, and special data structures are used to reduce state transition matrix, iteration vector, etc. (Sanders, W. H. et al. 1998).
For stiff models, with tolerance approach (STA) the *specialized numerical methods* (Arushanyan, O. et al. 1990, Reibman, A. et al. 1988) are used to provide highly accurate results despite stiffness. The limitations of STA are:

- STA cannot deal effectively with large models;
- 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 MM to a non-stiff chain first, which typically has a significantly smaller state space [6]. An advantage of this approach is that it can deal effectively with large stiff MMs.
Availability and safety assessment using MM for such complex systems as NPP I&C system can be assessed through model-based evaluation in specialized tools (λPredict, Möbius, SHARP, etc.), off-the-shelf tools (Maple, Matlab, Mathematica, etc.) and own developed utilities (ODU).
The variety of tools and techniques (T&T) 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 MMs 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

(Hairer, E. et al. 2010), and tools. This view goes against the recommendations of IEC 61508-6 (Standard IEC 61508 2010). This standard asserts that methods for solving MM have been developed long ago and trying to improve these methods does not seem sensible. The previous works show (Kharchenko, V. et al. 2013) 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 (Kharchenko, V. et al. 2013), thus, use of inappropriate method/tool for the solution of a non-trivial MM may lead to significant errors. One of the leading standards in the safety area IEC 61508-2010 provides no special requirements for 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) (Standard IEC 60300-3-3 Ed.2.0 (2008). In contrast to the T&T, many requirements were developed for I&Cs verification and validation tools are compatible by strength to the requirements of produced software and systems (see standard IEC 60880-2006 (Standard IEC 60800 Ed. 2.0 2006).

MAIN FOCUS OF THE CHAPTER

In this chapter presents 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 (Standard IEC 61508 2010), IEC 60300:2003 (Standard IEC 60300-3-3 Ed.2.0 2008), IEC 61165:2006 (Standard IEC 61165 2006) and ECSS-Q-ST-30-09:2008 (Standard ECSS-Q-ST-30-09 2008). By the "*gap*" we define special risks, which are accompanies the MC application procedure.
We discuss the ways how MM assessment risks and difficulties 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.

GAPS OF THE IEC 61508

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.
Figure 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.
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) (Standard IEC 61508 2010) and IEC 61165 (Standard IEC 61165 2006) assumes constant time-independent state transition rates, while in practice those rates can vary over time due to physical aging errors, etc.

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).

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 size.	IEC 61508 – 6 “…The main problem with Markov graphs is that the number of states increases exponentially when the number of 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 statistics 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 (Kharchenko, V. et al. 2013). 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 4. 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 the model manual construction, it is important to verify the resulting MC to detect the human-based errors.	IEC 61508 – 6 "…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 5. Gaps of the verification phase

Gaps	Reviewed 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 do 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.

DIFFICULTIES OF MARKOV MODELS APPLICATION

Stiffness

There is no common definition of “stiffness” because of its complexity. In publication (Malhotra, M. et al 1994) authors introduce the “practical” definitions of stiff problems, based on the interpretation of physical processes in research systems. Here we provide the example of “practical” formation of stiffness problem.

In general there are two ways of how to improve the availability: increase time-to-failure or reduce time-to-recovery. The system failures can be caused by various types of defects (bohrbugs, heisenbugs, aging-related bugs (Nicola, V.F. 1982) the rate of which may vary in orders (more then 10^2). The difference in orders of software – hardware system failure and recovery rates values (Kharchenko, V. et al. 2013) can be shown as an example that system has a feature of *stiffness* (Arushanyan, O. et al. 1990). The given difference appears in matrix of coefficients of Kolmogorov DE and lead to inefficiency of explicit numerical methods use (Malhotra, M. et al 1994).

In research works (Reibman A. et al. 1989, Hayrer, E. et a;. 1999) authors present the definition of stiffness based on the problems of numerical solution: inability or ineffectively use of explicit numerical methods; presence of quick perturbations decay; big Lipschitz constants; big difference of Jacobi matrix eigenvalues etc.

One of the most vides used stiffness definition a method is based on the calculation of stiffness index – s (Hayrer, E. et a;. 1999).

The Cauchy problem $\frac{du}{dx} = F(x,u)$ is said to be stiff on the interval [x_0, X] if for *x* from this interval the next condition is fulfilled:

$$s(x) = \frac{\max_{i=1,n} |\mathrm{Re}(\lambda_i)|}{\min_{i=1,n} |\mathrm{Re}(\lambda_i)|} ? \ \ 1, . \qquad (1)$$

where: s(x) – denotes the index of stiffness, λ_i – are the eigenvalues of a Jacobi matrix.

In work (Hayrer, E. et a;. 1999)] Ernst Hairer and Gerhard Wanner propose two possible methods of prior detection of stiffness in researches DE system. The implementation of automatically stiffness detection can help to avoid the not accurate, in case of stiffness, numerical methods. The first method is based on the analysis of errors on first steps of system DE solution (not more than 15 steps). The second possibility is based on the estimation directly the dominant eigenvalue of the Jacobian of the problem.

In the last 30 years a lot of approaches have been developed to deal with the problem of stiffness (Arushanyan, O. et al. 1990, Reibman A. et al. 1989). They can be separated into two groups - *stiffness-tolerance* (STA) and *stiffness-avoidance* approaches (SAA).

The basic idea of STA is a model transformation by identifying and eliminating the stiffness from the model, which would bring two benefits: a reduction of the largeness of the initial MM; 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 (Bobbio, A. et al. 1986), 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.
The main idea of SAA 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.
Detailed analysis of given classification attribute can determine the type of *stiffness* and according to it choose the optimal (as combination of estimation time, resource cost and accuracy) computational method (Bobbio, A. et al. 1986, Arushanyan, O. et al. 1990).
Based on the analysis of earlier research works (Kharchenko, V. et al. 2013, Butenko, V. 2014) and conducted empirical tests for each group of *s(x)* were selected the main MM solution technique and technique for results verification (Kharchenko, V., et. al. 2014). The Fig.2 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.

Figure 2. Normalized scale for stiffness metric

The top values on scale (Fig. 2) 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:

$$m_i = \frac{x_i - x_i^{max}}{x_i^{max} - x_i^{min}}, . \qquad (2)$$

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)* $x^{min} = 0$.

Largeness (Decomposability, Irreducibility)

As a second classification attribute of researched system model the term of *largeness* can be used.
In the modeling process the real object is presented with some level of specification. Determine the level of specification at different stages of the modeling process is unique for each system. The nowadays RTS are complex hardware-software systems. High requirements to the reliability of such systems operating process force the modeler to decompose the system to the elementary parts to provide the accurate in-depth analysis. The process of including more details in model makes it larger and more complicated so its analysis will be more difficult or even intractable.
Methods of large MM solution can be divided into two types: largeness-avoidance and largeness-tolerance.
The main idea of largeness-avoidance approach (LAA) 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 (Sanders, W. H., et al. 2003). The state-level and model-level (Sanders, W. H., et al. 2003) lumping techniques are well-known methods of LAA approach. A state-level lumping technique is a technique that exploits the certain properties on the MM level, while the model-level lumping denotes the lumping properties on the high-level formalism and directly construct lumped MM. 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 MM then a lumping technique.
The largeness-tolerance approach (LTA) is designed to manipulate large MM using special algorithms and data structures to reduce and store transition probabilities matrix (Srinivasan, A., et al. 1990). The numerous works (Srinivasan, A., et al. 1990, Bryant, R. E 1986) 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 (Sanders, W. H. et al. 1998, Gail, H. R., et al. 1989). Analysis of MM irreducibility and decomposability properties can help to make a prior selection between described techniques (Bryant, R. E 1986).
The analysis of MM using classification attribute largeness will reduce the time for system assessment using special algorithms and amount of computing resources.
The aggregation techniques are mainly based on the decomposability approach. In this case the degree of coupling can be taken as measure of matrix largeness (decomposability) property.
For example, considering a nearly completely decomposable (NCD) MM, which has a matrix with non-zero elements in off-diagonal blocks are small compared with those in the diagonal blocks (Courtois, P. J. 1977):

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

$A_{11}, A_{12}, \dots, A_{nn}$ 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 (4), where E contains all of-diagonal blocks. The quantity (5) is referred to as degree of decomposability (Courtois, P. J. 1977). If $E = 0$ then MM is said to be completely decomposable (CD).

$$A = diag(A_{11}, A_{22}, \dots, A_{nn}) + E, \tag{4}$$

$$\|E\|_{\infty} = \max_{1 \le i \le n} \sum_{j=1}^{n} |e_{ij}| \tag{5}$$

An irreducible MM 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 (Cormen, T. 2001). 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 (5) can help in deciding between avoidance and tolerance approaches, thus we refer to (5) as a main largeness metric, further decomposability metric. The E values can be split into three groups (Kharchenko, V., et. al. 2014):

- 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. 3 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 $x_{min} = 0$.

Figure 3. Normalized scale for decomposability metric

Sparsity

Conceptually, sparsity (Barge, W. S., et al. 2002) corresponds to systems which are loosely coupled. In the subfield of numerical analysis, a sparse matrix is a matrix populated primarily with zeros (Press, W.H. et al. 2007).
The analysis of classification attribute sparsity is important part for the special class of problems. As an example of such class we can use the solution of Kolmogorov DE, which describes the MM of system under research. As the matrix of DE coefficients is presented in mostly diagonal form so given attribute can be accompanying in case of using the apparatus of Markov modeling. If research MM is large the sparsity can cause additional assessment difficulties.
Storing and manipulating sparse matrices on a computer is beneficial and often necessary to use specialized algorithms and data structures that take advantage of the sparse structure of the matrix. Operations using standard dense-matrix structures and algorithms are relatively slow and consume large amounts of memory when applied to large sparse matrices.
Most of the approaches are developed to reduce the size of the transition matrix representation and form the dense matrix, by using structured analysis (Reibman, A. et al. 1988) or symbolic data structures analysis (Cormen, T. 2001) and solving them using lumping algorithms or iterative techniques (Cormen, T. 2001).
Paper (Barge, W. S., et al. 2002) presents the formula for evaluation of the heuristic measure of sparsity – matrix score. It gives a measure of how the matrix elements are dispersed from the main diagonal. The analysis of this score in the process of system research will reduce the time for system assessment by using specialized techniques, algorithms and data structures that take advantage of the sparse structure of the matrix.
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:

$$ms = \left(\sum_{i=1}^{n-1} i \cdot q_i\right) / n^2 \cdot \tag{6}$$

In (Barge, W. S., et al. 2002) 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 $n \geq 500$ 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 (Kharchenko, V., et. al. 2014):

- high sparsity: $ms < 0.3$;
- moderate sparsity: $0.3 \leq ms < 0.72$;
- low sparsity: $ms \geq 0.72$.

Fig. 6 presents normalized scale of ms with recommendations for approach selection, where $x_{max} = 0.9$ and $x_{min} = 0$.

Figure 4. Normalized scale for sparsity metric

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 (Kharchenko, V. et al. 2011). The verification and validation phases are strongly managed by requirements and recommendations of international standards (see standards IEC 60800-2006, IEC 61508 - 2010). There are lots of required procedures to test the software component, such as documentation analysis, problem review, static code analysis, etc (Butenko, V. 2014). 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 (Butenko, V. 2014, Kharchenko, V., et. al. 2014) we applied the *multi-fragmentation principle* to present such complex interconnection between system hardware and software. The main idea of this approach is to capture and represent using MC the plausible phenomenon – variation of software failure – that is well accepted on practice (Kharchenko, V. et al. 2013). 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 use of fragments in MM will increase the modeling clarity and take into account some properties of operating system modes. It is necessary to understand that introduction of parameters change assumption can increase the system size (direct affect on the feature *largeness*) and as a result the increase the sparsity of system transitions matrix (affect on the feature *sparsity*).

The number of fragments N_{fr} in MM depends on the number of expected undetected software faults n_i in i-different software versions:

$$N_{fr} = \prod_{i=1}^{m} (n_i + 1). \tag{7}$$

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 MM can be classified as follows:

- low-fragmented: $N_{fr} < 6$;
- moderately fragmented: $6 \leq N_{fr} < 15$;
- highly fragmented: $N_{fr} \geq 15$.

The normalized scale ($x^{max} = 30$ and $x^{min} = 0$) with recommendations for approach selection is presented on Fig. 5.

Figure 5. Scale for fragmentedness metric

METRIC-BASED DIAGRAM

The recommendations for selecting the solution approach presented on Fig. 2 – 5 were received separately for each characteristic. However, it is important to consider each MM feature, while making decision on the most efficient technique.

In this section we present the metric-based diagram (Fig. 6), that incorporates all MM 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.

Figure 6. Metric-based diagram

The diagram is applied in three main stages.

1. Calculation of stiffness, decomposability, sparsity and fragmentedness metrics using (1), (5) – (8).
2. Normalization of the received values using (2).
3. 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 cij, $i \in (\overline{1,4})$, $j \in (\overline{1,5})$. 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.

MSMC tool

The "MSMC – Method selector for Markov chains" was developed to help user, unsophisticated in Markov modeling to select the most effective solution approach automatically, which will give an accurate result. The previously described metric-based approach was implemented in MSMC. This is an alpha version, which runs on Windows platform (XP and later).
MSMC supports:

1. Two types of MM construction: graphical and analytical.
2. Stiffness, sparsity, decomposability and fragmentedness testing.
3. Selection of the solution approach using the tests results.
4. State probabilities calculation using selected approach.
5. Conversion of transition probabilities matrix and DES into the MATLAB, Mathematica and Maple syntaxes in verification purpose.

The major components of a GUI are model editor, transitions editor, generated matrix tab, solution tab and methods panel. The model editor (Fig. 7) allows a graphical construction of the MM using MSMC primitives – places (states) and arcs (transitions). If MM already exists user can upload the transition probabilities matrix as an Excel spreadsheet (.xls or .xslx) or text (.txt) document. With the transitions editor user can change already assigned transition values.
Figure 7. MM graphical construction

After the MM construction, MSMC generates an underlying transition probabilities matrix and see the result in generated matrix tab (Fig. 8). With the Export option user can convert the received matrix into MATLAB, Mathematica and Maple form and save it as text file.
Figure 8. Generated transition probabilities matrix

The stiffness, decomposability, sparsity and fragmentedness test results, metric-based diagram and recommendations for approach selection are displayed in solution tab (Fig. 9).
Figure 9. Metric-based approach application

If the matrix appears to be moderately or highly sparse MSMC will store the non-zero elements in flat format using the algorithm described in (Press, W.H. et al. 2007). Thus, all subsequences of calculations are performed on the compressed matrix form. We have implemented the well-known numerical methods (Press, W.H. et al. 2007) (implicit RK, explicit RK, exponential, etc.) and aggregation techniques (Bobbio, A. et al. 1986) to find the MC transient solution. Using the methods panel user can select the recommended method (-s), which are highlighted with color. The solution results are automatically saved into the text file.

ANALYSIS AND SELECTION OF TOOLS APPLICABLE DURING MARKOV MODELING

The IEC 60300-3-1:2003 states a list of questions, which helps to choose an appropriate method as well as software tool (ST), that can be applied during complex systems safety and reliability and parameters assessment. The main questions consider software usability concept and ability to reuse resulting data. The Table 6 presents comparison analysis of several tools which can be used during MM application – SHARP (Duke University), ToolKit Markov (ITEM), MARCA, MÖBIUS (University of Illinois Urbana Champaign), ASNA (Lviv Polytechnic National University), MSMC, Mathematica (Wolfram), Matlab (Mathworks) and Maple (Maplesoft).
Based on comparison of ST features for construction, analysis, editing, solution end exporting of the MM analysis results we have developed an algorithm of software tools selection and application (Fig. 10).
The Fig. 10 uses following abbreviations: *Ap* – approaches for MM analysis (8), *TA* – tolerance approaches (9), *AA* – avoidance approaches (10), *STA* – stiffness tolerance approach, *LTA* – largeness tolerance approach, *SAA* – stiffness avoidance approach, *LAA* – largeness avoidance approach, *T* – ST applicable during MM (11), *Build* – ST for MM construction (12), *Gen* – ST for MM generation, *BuildGr* – ST for MM graphical construction, *BuildTxt* – ST for MM text-based construction, *Analyze* – ST for metric-based analysis, *Sol* – ST for numerical MM analysis.

$$AA = \{TA, AA\} \tag{8}$$

$$AA = \{SAA, LAA\} \tag{9}$$

$$AA = \{SAA, LAA\} \tag{10}$$

$$T = \{Build, Gen, Analyze, Sol\} \tag{11}$$

$$Build = \{BuildGr, BuildTxt, Gen\} \tag{12}$$

Table 6. Feature-based comparison of tools applicable during Markov modeling

Features	Tools								
	SHARP	Tool Kit Markov	MARCA	Möbius	ASNA	MSMC	Mathematica	Matlab	Maple
MM Construction									
Manually (graphic)	+	+				+			
Manually (text)	+		+				+	+	+
State-space generation			+	+	+				
Importing matrix of DE coefficients									
Import from text file	+		+			+	+	+	+
Import from table editor						+			
Metric-based analysis									
Detection of stiffness						+	+		+
Detection of largeness						+	+	+	+

Detection of sparsity			+			+			
Detection of periodicity			+			+	+		
Numerical methods									
For steady-state analysis	+	+	+	+		+	+	+	+
For transient analysis	+	+	+	+		+	+	+	+
Reports									
Reports generation	+	+	+	+	+	+			
Results export to the text or table editors		+	+			+	+	+	+

The algorithm can be divided into following logical parts:

1. *Blocks 1 – 4: Initial data.* The set of ST (11) and approaches (8), requirements to the accuracy, modeling time interval, acceptable error and required algorithm complexity – are the initial data for presented algorithm.
2. *Blocks 5 – 13: MM construction*. Based on data gathered after technical documentation analysis, assumptions on how deep the modeled system behavior should be presented during modeling process we can get the initial understanding of MM complexity. In case if resulting MM can be classified as large and complex the most appropriate way is to use automatic state-space generation instead of creating it manually. To do so, researcher can use various high-level modeling formalisms, such as stochastic activity networks, stochastic Petri nets of even verbal presentation. In other case the MM can be created manually with graphical or text editors. As the result researcher should obtain MM and Kolmogorov DE system that describes it.
3. *Block 14: Export of the matrix of DE coefficients into text editor.* This stage is required to perform further metric-based analysis of MM.
4. *Block 15 – 16: Metric-based analysis of the MM.* This stage includes application of MSMC methods or several tools which allow user to perform stiffness, largeness, sparsity and periodicity (fragmentedness) analysis. The result of block 16 is applied to make initial recommendations for approach selection.
5. *Block 17 – 25: Numerical MM analysis.* Based on results of metric-based analysis we can further perform selection of ST that includes the recommended approach. This stage also consider the required accuracy level – if it exceeds 10^{-6} the researcher is recommended to verified obtained numerical results.
6. *Block 26: Results analysis*. The obtained result are further analyzed by researcher to make informed decisions on modeled system behavior and properties.

Figure 10. Software selection and application during MM

MULTI-FRAGMENTARY MARKOV MODELING FOR THE REACTOR TRIP SYSTEM: CASE STUDY

RadICS based reactor trip system

This section presents the description of a studied 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 analogue 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.

Each track, observed in this paper, consists of five modules: LM, DIM, DOM, AIM and AOM. The Fig. 11 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.
Figure 11. The structure diagram of a typical track

The RTS reliability-block diagram is presented on Fig. 12. Reliability index Ppfi.j determines hardware reliability of the track Ti.j (defined by physical faults), where i indicates main (T1.j) or diverse (T2.j) channels, and j indicates the number of the track. Reliability index Pdfi determines software reliability of the main or diverse channels (defined by software faults), where i indicates channel. Reliability index P_{mi}, determines reliability of the majority element mi, where $i \in (\overline{1,3})$.
Figure 12. Reliability-block diagram of two-channel three-track system

Informally, the system operates as follows. Initially, all components (tracks) are working correctly and deliver the service as expected. If one of the tracks in main channel has failed, by majority voting the failure of the failed track is detected and while the failed track is being repaired, the operation will continue with the second and third tracks in channel. If before the first track has been repaired another one fails, the majority voting component will detect the output disagreement and main channel will stop. The diverse channel operates identically to the main. The operation process continues until one of channels works correctly. Clearly, the reliability of the majority components affects significantly the system reliability. In the observed RTS architecture the failure of majority components (P_{m1} or P_{m2}) leads to the failure of the main or diverse channels, and to the general RTS failure (P_{m3}).

Development of RTS Markov model and selection of the tool

Let us further denote by λd and λp the design and physical failure rates, respectively, and by μ_d and μ_p the repair rates after design and physical failures. The MM for RTS system is shown on Fig. 13.
Figure 13. Multi-fragmented Markov model of RTS

The detailed model construction and description is shown in (Butenko, V. 2014). We use the following assumptions to create the MM for observed RTS:

1. Each element of the research system in random moment of time can be only in two states – working and failure.
2. The systems control and majority elements provide unstoppable correct functioning.
3. The system maintenance is performed by one group of engineers, thus failed chassis are repaired sequentially. It should be noted, that recovering strategy use case of two working channels in a priority.
4. 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 (Butenko, V. 2014).
5. Software testing datasets are updated after each test. The testing is performed on the complete body of input data.
6. 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 (Butenko, V. 2014, Ehrlich, W., et al. 1990).
7. The failure rate of the design faults $\lambda_{d(i)}$ is proportional to their residual amount n_i in i – different software versions (Butenko, V. 2014). 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 (Kharchenko, V. et al. 2011).

8. 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:

$$\lambda_{d1} = \lambda_{d2} \Rightarrow \lambda_d = \lambda_{d1} + \lambda_{d2}, . \tag{13}$$

$$\mu_{d1} = \mu_{d2}; \mu_d = \lambda_d / (\sum_{i=1}^{2} \frac{\lambda_{d(i)}}{\mu_{d(i)}}) . \tag{14}$$

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 7.

Table 7. MM parameters value

	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}	$5 \cdot 10^{-6}$	10^{-4}	0.01	1	[0; 30 000]
2	$2.5 \cdot 10^{-5}$	$1.25 \cdot 10^{-5}$				
3	$5 \cdot 10^{-5}$	$2.5 \cdot 10^{-5}$				
4	$7.5 \cdot 10^{-5}$	$3.75 \cdot 10^{-5}$				

Analysis of MM stiffness, largeness, sparsity and fragmentedness as well as a MM manual graphical construction was performed in MSMC tool. The result of metric-based diagram application is presented on Fig. 14.

1. Stiffness: moderately stiff with s(x) = 0.167.
2. Decomposability: completely decomposable (CD) with E = 0.02.
3. Sparsity: highly sparse with ms = 0.2.
4. Fragmentedness: moderately fragmented with Nfr=6.

With the metric-based diagram (Fig. 10), 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).

Figure 14. Metric-based diagram for RTS

MODELING RESULTS

We use the recommended approaches to assess the RTS unavailability function, which also defines the PFH measure [11].

The unavailability function U(t) is defined as a sum of failed states probabilities, with initial condition U(0) = 0:

$$U(t) = 1 - A(t) = \sum_{i=1}^{n} P_i(t), i \in N, . \tag{15}$$

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

Due to recommendations from metric-based approach (Fig . 14) and application of tools selection algorithm (Fig. 10) we use the STA as a main techniques, thus the *U(t)* for four sets of RTS parameter (Table 7) was calculated using build-in function of implicit Runge-Kutta (RK) method in Mathematica. The results are shown on Fig. 15.
Figure 15. U(t) calculated using STA

The *U(t)* result were verified using MSMC inner function, that implements the implicit RK algorithm and SAA, particularly aggregation/ disaggregation technique (Bobbio, A. et al. 1986). The result of *U(t)* verification for $\lambda_d = 5 \cdot 10^{-5}$ is presented on Fig. 16.
Figure 16. U(t) results verification

FUTURE RESEARCH DIRECTIONS

The future research work will consider the development of a metric-based algorithm with a self-improvement ability to generate more accurate recommendation for techniques selection.

CONCLUSIONS

There are some gaps in standards including IEC 61508 regarding techniques and tools choice and accuracy of indicators evaluation on MM-based assessment of safety-critical systems. We conclude that such difficulties as MM 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 MM; 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.
This chapter presents the metric-based approach for selection of the applicable solution approach, based on the analysis of such MMs characteristics as stiffness, largeness (decomposability, irreducibility), sparsity and fragmentedness. The metric-based approach was implemented into MSMC tool that support the graphical construction of MM, with further generation of underlying matrix and testing its properties. Based on the test results MSMC tool automatically presents the recommendations for approach selection and provides the transient solution using inner numerical methods.
We present the features-based comparison and discuss the way how software tools can be selected and applied during the modeling of complex systems using MM.
This chapter also presents the case study for assessment the unavailability parameter of NPP I&Cs, in particular, RTS. The system model was build using the MM, taking into account the failures caused by physical and design faults.

REFERENCES

Trivedi, K.S. et al. (2000). Availability Models in Practice. Proc. *Int. Workshop on Fault-Tolerant Control and Computing (FTCC-1)*, May 22-23, Seoul, Korea.
Malhotra, M. et al (1994). Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains, *Microelectronic Reliability, Vol.34(11),* 1825-1841.
Buchholz, P. (1996). Structured analysis approach for Large Markov Chains. A Tutorial. *Proc. of Performane'96,* Lausanne, Switzerland.
Hurley N. et al. (2009). Comparing measures of sparsity. *IEEE Trans. on Information Theory,* 2009, Vol.55, №10, 4723 – 4741.

Press, W.H. et al. (2007). *Numerical Recipes. The Art of Scientific Computing*, Cambridge University Press.
Bobbio, A. et al. (1986). An Aggregation Technique for Transient Analysis of Stiff Markov Chains, *IEEE Transactions on Computers*, 803-814.
Arushanyan, O. et al. (1990). *Numerical Solution of Ordinary Differential Equations using FORTRAN*, Moscow State University, Moscow.
Sanders, W. H. et al. (1998). An Efficient Disk-based Tool for Solving Large Markov Models, *Performance Evaluation, Vol. 33*, 67–84.
Reibman, A. et al. (1988). *Numerical Transient Analysis of Markov models*. Comput. Opns. Res., 15(1), 19-36
Hairer, E. et al. (2010). Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, *Springer*, 631.
Standard IEC 61508 (2010). *Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-Related Systems*.
Kharchenko, V. et al. (2013) *Availability Assessment of Computer Systems Described by Stiff Markov Chains: Case Study*, Springer, CCIS , Vol. 412, pp. 112 - 135.
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*.
Standard IEC 60300-3-3 Ed.2.0 (2008). *Application guide – Analysis techniques for dependability – Guide on methodology*.
Standard IEC 61165 (2006). *Application of Markov technqiues*.
Standard ECSS-Q-ST-30-09 (2008). *European Cooperation for Space Standardization (ECSS): Availability analysis*.
Nicola, V.F. (1982). *Markovian Models of Transactional System Supported by Check Pointing and Recovery Strategies, Part 1: a Model with State-Dependent Parameters*. Eindhoven Univ. Technol., Eindhoven, The Netherlands, EUT Rep. 82-E-128.
Reibman A. et al. (1989). *Analysis of Stiff Markov Chains*. ORSA Journal on Computing, 1(2), 126-133pp.
Hayrer, E. et a;. (1999). *Solution of Ordinary Differential Equations. Stiff and Differential-Algebraic Poblems*. Mir, Moscow.
Butenko, V. (2014). Modeling of a Reactor Trip System Using Markov Chains: Case Study, *Proc. of 2014 22nd ICONE*, Vol. 5.
Kharchenko, V., et. al. (2014). Markov's Model and *Tool-Based Assessment of Safety-Critical I&C Systems: Gaps of the IEC 61508, In* Proc. 12th Int. Conf. PSAM, 16.
Sanders, W. H., et al. (2003). Optimal State-space Lumping in Markov Chains, *Inf. Process. Lett., Vol. 87(6)*, 309-315.
Srinivasan, A., et al. (1990). Algorithms for Discrete Functions Manipulation, *In Proc. Int'l Conf. on CAD (ICCAD'90)*, 92-95.
Bryant, R. E (1986). Graph-based Algorithms for Boolean Function Manipulation, *IEEE Trans. Comp., Vol. 35(8)*, 677–691.
Gail, H. R., et al. (1989). Calculating Availability and Performability Measures of Repairable Computer Systems, *Journal of the ACM, Vol. 36*, 171–193.
Barge, W. S., et al. (2002). Autonomous Solution Methods for Large Markov Chains, *Pennsylvania State University CiteSeerX Archives*, 17.
Courtois, P. J. (1977). *Decomposability: Queueing and Computer Applications*, Academic Press, New York.
Cormen, T. (2001). *Introduction to Algorithms*, MIT Press, Computers, 180.
Kharchenko, V. et al. (2011). Multi-fragmental Availability Models of Critical Infrastructures with Variable Parameters of System Depend*ability*, *Int. Journal Information & Security*, Vol 28, 248 – 265.
Ehrlich, W., et al. (1990). Applying Reliability Measurement: A Case Study, *IEEE Software*, 1990, 56-64

ADDITIONAL READING SECTION

IAEA. (1999). *Modern Instrumentation and control for NPP: A guidebook*, No. 387.
Smith, W. E., et al. (2008). *Availability Analysis of Blade Server Systems*, IBM Systems Journal, Vol. 47(4), pp. 1- 20.
P. Buchholz, et al. (2004). *Approximate Computation of Transient Results for Large MC*, In proc. of IEEE QUEST'04, pp. 126-135.
Miettinen, K., et. al. (2008). *Multiobjective Optimization. Interactive and Evolutionary Approaches*, Springer-Verlag, Berlin, Heidelberg, p. 461.
V. Kharchenko, A. Siora, V. Sklyar et al. (2010). *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.

KEY TERMS & DEFINITIONS

Architecture: Specific configuration of hardware and software elements in a system.

Channel: Element or group of elements that independently implement an element safety function.

Safety Integrity Level (SIL): Discrete level (one out of a possible four), corresponding to a range of safety integrity values, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest.

Largeness: Structural complexity of the model.

Multi-fragment Markov model: Set of repetitive macromodels (fragments), the internal structure of which and the external relations depend on the sets of selected basic parameters of the model.

Sparsity: Sparse matrix is a matrix populated primarily with zeros.

Stiffness: Value of that in models of complex computer systems is caused by failure and recovery rates varying by several orders of magnitude.