MODELING THE RELIABILITY OF C4ISR SYSTEMS HARDWARE/SOFTWARE COMPONENTS USING AN IMPROVED MARKOV MODEL

Cezar VASILESCU

Regional Department of Defense Resources Management Studies, Brasov, Romania

Abstract*: An important problem for C4ISR systems is choosing the right reliability modeling method. Because C4ISR system's components are redundant and configurable, both at the level of functional entities and inside them, the resulting reliability models are complex. The redundancy adds supplementary complexity elements associated with the response capability of hardware and software to the failure events. A complicated issue is to choose the right degree of detail. Too many details could result in loosing the existing hardware and software interdependencies of the redundant structures within the model. In this paper I am proposing an enhanced theoretical calculus method of the reliability indicators based on an improved Markov model, and its application to the reliability modeling of C4ISR systems hardware/software components. A detailed application of the enhanced method is presented in paragraph 3. The reliability modeling of C4ISR systems' components may use status diagrams recommended for redundant hardware/software structures. Nevertheless, this solution could be hard to be applied, considering the C4ISR system's complexity, its number of possible states, and the large number of redundant elements and parameters of interest. Therefore I approach the study by adapting the status diagrams representation, through a combination of system's status tables with transitions states graphs, using also the matrix representation of fluency graphs.*

Keywords*: hardware/software systems, reliability analysis, dynamic models, Markov model.*

1. INTRODUCTION

C4ISR systems are basically computerized networks that include sensors, communication nodes and command centers, each of it performing special / essential tasks for good functioning of the entire system. From a reliability point of view it represents a serial reliability structure made of independent components, each with its own functional restore capabilities - ultimately a system with multiple recovery capacity.

Each of the system's components could have a more or less complex redundant structure, but we should consider it as a whole. Consequently the system could be considered a minimal serial structure with multiple recovery capacity. The considerations regarding the reliability modeling of combined hardware/software systems had as a starting point the Rome Laboratory's *"System and Software Reliability Assurance Notebook"* (Lakey & Neufelder, 1997). The document offers a methodology for the reliability assurance study of systems composed both from hardware and software elements, designed for estimation and prediction of its reliability. The methodology concentrates on software configuration reliability prediction and estimation methods and on the techniques used to correlate hardware and software reliability metrics with system's general parameters. The reliability model proposed by Rome Laboratory consists of state transition diagrams that partially respond to the issues related with the complexity of a C4ISR system and its components. Even if the model contains restrictive hypotheses, the reliability indicators of the system's components calculus can utilize a large scope of methods. Consequently, in the first part of the paper I am proposing an enhanced theoretical calculus method of the reliability indicators based on an improved Markov model, while in the second part I apply it to the reliability modeling of C4ISR systems hardware/software components.

A complicated issue was to choose the right degree of detail. Too many details could result in loosing the existing hardware and software interdependencies of the redundant structures within the model. Because the C4ISR system's components are redundant and configurable, both at the level of functional entities and inside them, the resulting reliability models are complex. The redundancy adds supplementary complexity elements associated with the response capability of hardware and software to the failure events. For the reliability modeling of C4ISR systems' components I used one of the general methods applicable to complex redundant structures, based on hardware, software and hardware/software elements - the Markov processes method. A detailed application of the enhanced method will be presented in paragraph 3. The reliability modeling of C4ISR systems' components may use the status diagrams recommended for redundant hardware/software structures.

Nevertheless, this solution could be hard to be applied, considering the C4ISR system's complexity, its number of possible states, the large number of redundant elements and parameters of interest (for only two identical elements and five parameters of interest it results 32 possible states). Therefore I approach the study by adapting the status diagrams representation proposed by Rome Laboratory, through a combination of system's status tables with transitions states graphs, using also the matrix representation of fluency graphs.

2. MODELING THE SYSTEM'S COMPONENTS RELIABILITY USING AN IMPROVED MARKOV METHOD

The reliability modeling process of system's components through Markov processes method uses a dynamic system model (Ghi□ă & Ionescu, 1996:55-56). These dynamic models follow the temporal evolution of the system's state and are defined using characteristic states. A system's state represents synthetic information regarding the condition of its elements and could have only two possible values: an operational state or a failure state. Assuming n is the number of elements of the system and X the set of system's states $X = \{x_i; i = 1,2,...,N\}$, in which $N \le 2^n$, it results that X is a finite set with elements defined as $x_i = (x_{i1}, x_{i2},..., x_{in})$ and $x_{ik} = 0$ when element k is in an operational state and $x_{ik} = 1$ when element k is in a failure state, $k = 1,2,...,n$ for state x_i.

Due to the failure and recovery of component elements, the current state of the system at t moment, noted $\upsilon(t), t \in T = [0,\infty]$ is a random process. Mathematically speaking, this process has the following characteristics: evolves in a continuum time; has a finite set of states X and has a poissonian character (in a very short time interval it could be distinguished maximum one change of state).

The subset $S \subset X$ represents the set of failure states and the complementary subset $\overline{S} = X - S$ represents the set of operational states. The model allows to determine the time duration when the process $\upsilon(t)$ will remain in S or $\overline{S}$. A random process with the above mentioned characteristics could be modeled as a graph $G = (X, \Lambda)$, where X represents the nodes set (identified with the states set) and Λ represents the set of arches that connect the nodes.

Each poissonian process can be associated with a graph following certain rules (Lakey & Neufelder, 1997): the nodes identify the states; nodes are connected one to each other through arches if there is a possibility to move from one state to other in a single step; on arches is specified the reliability or maintainability parameter that determines the transition, λ_i for failures and μ_i for recoveries.

The resulting graph is named States Transition Graph. The Markov processes method is based on a dynamic model that follows the evolution in time of the current system's state $\upsilon(t)$. On each moment t, $\upsilon(t)$ is a random discrete variable. Let's note $p_k(t) = P(\upsilon(t) = x_k)$. The probabilities $p_k(t)$ are absolute probabilities with fundamental properties:

$$1)\ p_k(t) \geq 0$$

$$2)\ \sum_{k=1}^{N} p_k(t) = 1 \qquad (1)$$

Poissonian processes theory shows that those absolute probabilities are solutions for a differential equation system that could be determined by applying Venttel rule to the state transition graph.

Derivative $p_k^{'}(t)$ is equal with the algebraic sum of the products between the starting nodes' probabilities and the corresponding transition intensities of the arches leaving those nodes, calculated for all incoming and outgoing arches belonging to node x_k. In the sum, the terms entering x_k are positive $(+)$, while the exiting ones are negative $(-)$.

Let's note Λ_k^{+} the set of arches entering node k, Λ_k^{-} the set of arches exiting node k and λ_{ij} the transition intensity of the arch that exit node i and enter node j. The following equations system results:

$$p_k^{'}(t) = \sum_{\lambda_{jk} \in \Lambda_k^{+}} p_j(t)\lambda_{jk} - \sum_{\lambda_{kj} \in \Lambda_k^{-}} p_k(t)\lambda_{kj},\ k = 1, 2, ..., N \qquad (2)$$

The system of equation is integrated for the initial condition (1):

$$p_1(0) = 1$$

$$p_k(0) = 0,\ k \neq 1.$$

where $x_1 = (0,0,...,0)$ is the state in which all elements are operational.

The main reliability indicator of a system calculated through this method is the probability function $P(t)$. Having $\bar{S} \subset X$ - the set of failure states and S - the set of operational states, it results:

$$P(t) = \sum_{k \in \bar{S}} p_k(t) \qquad (3)$$

In order to calculate this probability it is necessary to eliminate from the state transitions graph all the transitions from $\bar{S}$ to S. If the graph maintains his structure complete, through the same relation we can obtain the system's availability coefficient $K_D(t)$ or $D(t)$. It should be mentioned that when we apply the method to C4ISR systems, the following indicators must be calculated:

$$F_{def(i)}(t) = 1 - P(t) \qquad (4)$$

and

$$T_{def(i)} = \int_0^{\infty} P(t)dt. \qquad (5)$$

The advantage of the method is the transformation of system's reliability calculus to standard mathematical problems. It applies both irreparable and repairable systems, when failure and restore follow an exponential distribution law. The method implies a great volume of calculations which exponentially increases with the system's states number.

Applying the method to C4ISR systems needs data regarding the software failure rates (operating systems of applicative software) and hardware failure rates. The operating systems' failure rates could be obtained from the manufacturers, usually as number of failures in a specific time interval.

Those failure rates are calculated taking in consideration the time of the system, because the operating system is active when the computer is on and ready for processing. To obtain the compatibility with hardware failure rates the measuring unit of the obtained failure rates must be numbers of failures per hour.

Regarding the failures of reusable software code, that information could be obtained from the applications in which the software was used before. The information is relevant only if the operational profile of the previous designed application is approximately the same with the current application and in this case the historical data regarding failure rates could be used.

In comparison with Rome Laboratory's approach, reliability analysis with dynamic models has *several advantages*, such as:

- systematize the states transition graph, because the arches connect only the states that differ one from each other with a single defect;
- offers a systematization surplus, the status diagrams (states number) are organized tacking in consideration the number of existing system's failures;
- offers the possibility to represent the states transition graph using fluency graphs methods (incident / adjacent matrix) and
- offers a complete methodology for the calculus of reliability indicators.

3. IMPLEMENTATION OF THE RELIABILITY MODELING METHOD TO C4ISR SYSTEMS HARDWARE/ SOFTWARE COMPONENTS. CASE STUDY

The objective of the paragraph is to detail the implementation of the improved Markov methodology presented before to the reliability modeling of C4ISR systems. For this purpose we will use the reliability model of a simple redundant structure. Figure 1 depicts the simplified state diagram of a hardware/software structure with two identical elements (at least one must be in an operational state). To model the system's states, I took into account five parameters: status of the basic hardware platform (operational or faulty); status of the basic software (operational or faulty); status of the backup hardware platform; status of the backup software and status of the operation's recovery.

System's operation recovery may be successful or fail. Success is defined by: the structure switched on backup equipments, as a result of a basic hardware or software failure; a failure to the backup equipment has been detected, and it has been repaired.

Because there are five parameters of interest, each with two possible states, it results a total of 32 states. Nevertheless, in practice, some of the states could not exist (e.g. software could not be operational on a faulty hardware platform) and others are functionally equivalent.

The following notations were used: # - state number; T - type (success or failure); A - basic hardware; B - basic software; C - backup hardware; D - backup software; E – recovery, where A, B, C, D values are O = operational; F = failure; - = not applicable; E values are S= successful recovery or failure on backup equipment detected; F - unsuccessful recovery or latent failure on backup equipment.

The structure described has a total number of states equal with 10, as shown in table 1.

Table 1. Description of system's states

State no.	State type	Explanations
0	Success	Complete operational structure
1	Success	A software failure was detected in the backup equipment that was repaired
2	Success	The system is operational, with a latent software failure in the backup equipments
3	Success	The system is operational, with a detected hardware failure in the backup equipments
4	Success	The system is operational, with a latent hardware failure in the backup equipments
5	Failure	Failure on basic software, switching on backup hardware and software equipments failed; an intervention is necessary for system's recovery on either of hardware platforms
6	Failure	Failure on basic hardware and the recovery process failed; it occurred an incorrect detection, isolation or incomplete recovery; a manual intervention is required for system's recovery on backup equipments
7	Failure	Software failures occurred on both basic and backup equipments
8	Failure	Basic hardware and backup software failed; recovery is not possible without a manual intervention
9	Failure	Both hardware equipments failed

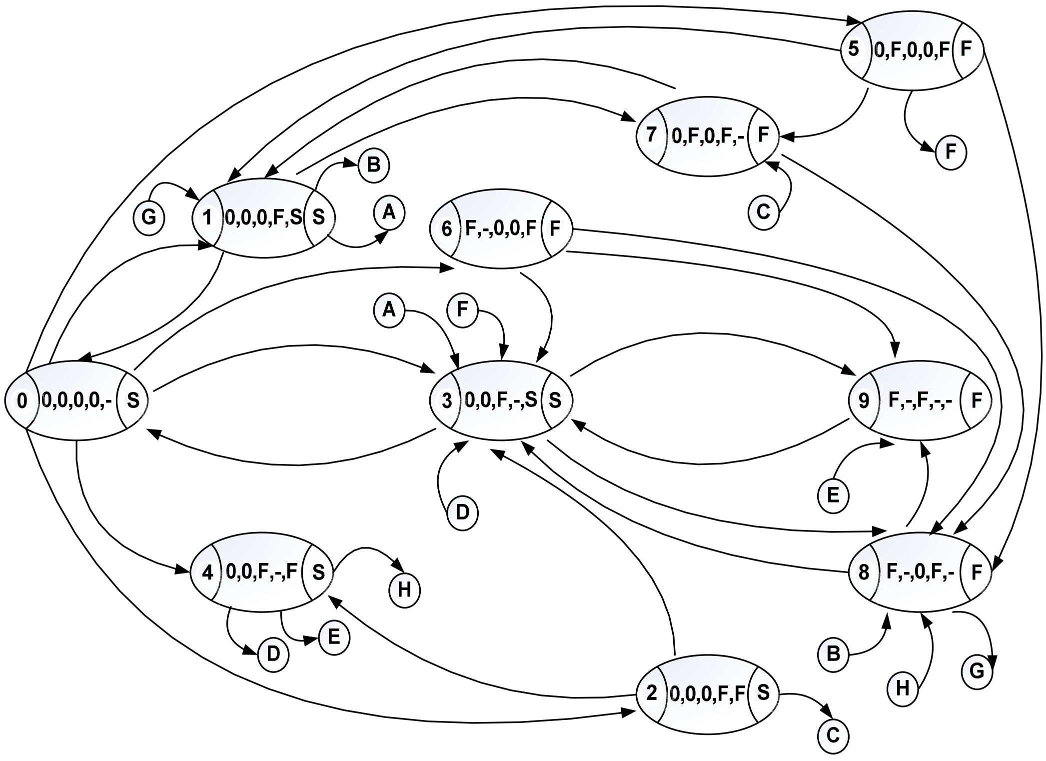

Fig.1. The simplified state diagram

Transitions from one state to another take place due to hardware (software) failures or as a consequence of recovery process. Usage of state diagrams that model hardware and software failures permit a precise redundant systems' reliability estimation.

Comparing these two approaches, it result some *observations*, such as:

1. The methodology for modeling the system's components reliability using an improved Markov method (presented in paragraph 2) is a applicable to every type of system (hardware, software or hardware/software) and allow the calculus of reliability indicators relevant for C4ISR systems (e.g. ($P(t)$ or $D(t)$, respectively D_S). The main problem related with its implementation is determined by the state transition graph's implementation manner and especially by the states' transition rates.

2. The states diagram presenting manner (figure 1) is similar with the states representation manner from paragraph 1, completed with specific aspects regarding the recovery of hardware/software system's components. Moreover, it introduces also state diagram systematization, presenting a list of states' transition rates (table 3).

3. The implementation of the reliability modeling method is conditioned by the knowledge of numeric values of probabilities such as C_{dSW} , C_{dHW} , C_{iSW} , C_{iHW} , C_{rSW} and C_{rHW} that must be inferred analytically and/or statistically,

where:

- *Fault detection coverage* (C_d) - is the probability of detecting a fault given that a fault has occurred;
- *Fault isolation coverage* (C_i) - is the probability that a fault will be correctly isolated to the recoverable interface (level at which redundancy is available) given that a fault has occurred and been detected and
- *Fault recovery coverage* (C_r) - is the probability that the redundant structure will recover system services given that a fault has occurred, been detected, and correctly isolated.

To facilitate the implementation, it results three new architectural products, specific to the C4ISR system's Reliability Vision (Vasilescu, 2010:198-200), as follows: **RV-6 -**

Description of states (table 2); **RV-7** - *The matrix of states' transition rates* (table 3); **RV-8** - *The list of states' transition rates* (table 4).

Table 2 derives from the simplified state diagram (fig.1).

Table 2. Description of states

Composition Index	A	B	C	D	E	Meaning
0	O	O	O	O	-	S
1	O	O	O	F	S	S
2	O	O	O	F	F	S
3	O	O	F	-	S	S
4	O	O	F	-	F	S
5	O	F	O	O	F	F
6	F	-	O	O	F	F
7	O	F	O	F	-	F
8	F	O	O	F	-	F
9	F	-	F	-	-	F

Through the table we divide the states in S and F, their index starting with working states (S) - state $(O, O, O, O, -)$, continuing step by step with every change of state generated by its corresponding elements. In this representation, all failure states (F) are shown in the lower part of the table.

Table 3. The matrix of states' transition rates

Index	0	1	2	3	4
0		(0, 1)	(0, 2)	(0, 3)	(0, 4)
1	(1, 0)			(1, 3)	
2				(2, 3)	(2, 4)
3	(3, 0)				
4				(4, 3)	
5		(5, 1)		(5, 3)	
6				(6, 3)	
7		(7, 1)			
8		(8, 1)		(8, 3)	
9				(9, 3)	

Index	5	6	7	8	9
0	(0, 5)	(0, 6)			
1			(1, 7)	(1, 8)	
2			(2, 7)	(2, 8)	
3				(3, 8)	(3, 9)
4				(4, 8)	(4, 9)
5			(5, 7)	(5, 8)	
6				(6, 8)	(6, 9)
7				(7, 8)	
8					(8, 9)
9					

In this matrix (Λ), the lines contain states' transition rates corresponding to outgoing arches form the node and columns contain states' transition rates corresponding to incoming arches form the node (essential information for the associated differential equation system).

Table 4. The list of states' transition rates.

Code	Content
(0, 1)	$\lambda_{SW} * C_{dSW} + \lambda_S * C_{dSW} * C_{iSW} * C_{rSW}$
(0, 2)	$\lambda_{SW}(1 - C_{dSW})$
(0, 3)	$\lambda_{HW} * C_{dHW} + \lambda_H * C_{dHW} * C_{iHW} * C_{rHW}$
(0, 4)	$\lambda_{HW}(1 - C_{dHW})$
(0, 5)	$\lambda_{SW}(1 - C_{dSW} * C_{iSW} * C_{rSW})$
(0, 6)	$\lambda_{HW}(1 - C_{dHW} * C_{iHW} * C_{rHW})$
(1, 0)	μ_{SW}
(1, 3)	λ_{HW}
(1, 7)	λ_{SW}
(1, 8)	λ_{HW}
(2, 3)	$\lambda_{HW} * C_{dHW}$
(2, 4)	$\lambda_{HW}(1 - C_{dHW})$
(2, 7)	λ_{SW}
(2, 8)	λ_{HW}
(3, 0)	μ_{HW}
(3, 8)	λ_{SW}
(3, 9)	λ_{HW}
(4, 3)	$\lambda_{HW} * C_{dHW}$
(4, 8)	λ_{SW}
(4, 9)	λ_{HW}
(5, 1)	$(Tr_{SW})^{-1}$
(5, 3)	λ_{HW}

Code	Content
(5, 7)	λ_{SW}
(5, 8)	λ_{HW}
(6, 3)	$(Tr_{HW})^{-1}$
(6, 8)	λ_{SW}
(6, 9)	λ_{HW}
(7, 1)	μ_{SW}
(7, 8)	$2\lambda_{HW}$
(8, 1)	μ_{HW}
(8, 3)	μ_{SW}
(8, 9)	λ_{HW}

We also observe that the matrix (Λ) has a cellular format (marked in table 3)

$$\Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}, \qquad (6)$$

in which Λ_{11} contains transition rates between working states (S), Λ_{12} contains transition rates from working states (S) to failure states (F), Λ_{21} contains transition rates from failure states (F) to working states (S), Λ_{22} contains transition rates between failure states (F). To calculate the availability (D) we use the matrix

$$\Lambda_D = \Lambda \qquad (7)$$

and to calculate the reliability (P) we use the matrix

$$\Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ 0 & \Lambda_{22} \end{pmatrix} \qquad (8)$$

in which 0 is the null identical matrix, when the determination methodology of the differential equation system associated with the model is implemented.

It is remarkable that this approach significantly orders (systematizes) the deduction of the differential equation system associated with the model (Gupta, 2011; Lai *et al.*, 2002). Thus to calculate the availability (D) it result the following systems of equations:

$$\begin{cases} p_0' = -[(0,1)+(0,2)+(0,3)+(0,4)+(0,5)+(0,6)]p_0 + (1,0)p_1 + (3,0)p_3 \\ p_1' = (0,1)p_0 - [(1,0)+(1,3)+(1,7)+(1,8)]p_1 + (5,1)p_5 + (7,1)p_7 + (8,1)p_8 \\ p_2' = (0,2)p_0 - [(2,4)+(2,3)+(2,7)]p_2 \\ p_3' = (0,3)p_0 + (1,3)p_1 + (2,3)p_2 - [(3,0)+(3,8)+(3,9)]p_3 + (4,3)p_4 + (5,3)p_5 + (6,3)p_6 + (8,3)p_8 + (9,3)p_9 \\ p_4' = (0,4)p_0 + (2,4)p_2 - [(4,3)+(4,8)+(4,9)]p_4 \\ p_5' = (0,5)p_0 - [(5,1)+(5,3)+(5,7)+(5,8)]p_5 \\ p_6' = (0,6)p_0 - [(6,3)+(6,8)+(6,9)]p_6 \\ p_7' = (1,7)p_1 + (2,7)p_2 + (5,7)p_5 - [(7,1)+(7,8)]p_7 \\ p_8' = (1,8)p_1 + (2,8)p_2 + (3,8)p_3 + (4,8)p_4 + (5,8)p_5 + (6,8)p_6 + (7,8)p_7 - [(8,1)+(8,3)+(8,9)]p_8 \\ p_9' = (3,9)p_3 + (4,9)p_4 + (6,9)p_6 + (8,9)p_8 - (9,3)p_9 \end{cases} \qquad (9)$$

while to calculate the reliability (P) it result the following systems of equations:

$$\begin{cases} p_0' = -[(0,1)+(0,2)+(0,3)+(0,4)+(0,5)+(0,6)]p_0 + (1,0)p_1 + (3,0)p_3 \\ p_1' = (0,1)p_0 - [(1,0)+(1,3)+(1,7)+(1,8)]p_1 \\ p_2' = (0,2)p_0 - [(2,4)+(2,3)+(2,7)]p_2 \\ p_3' = (0,3)p_0 + (1,3)p_1 + (2,3)p_2 - [(3,0)+(3,8)+(3,9)]p_3 + (4,3)p_4 \\ p_4' = (0,4)p_0 + (2,4)p_2 - [(4,3)+(4,8)+(4,9)]p_4 \\ p_5' = (0,5)p_0 - [(5,7)+(5,8)]p_5 \\ p_6' = (0,6)p_0 - [(6,8)+(6,9)]p_6 \\ p_7' = (1,7)p_1 + (2,7)p_2 + (5,7)p_5 - (7,8)p_7 \\ p_8' = (1,8)p_1 + (2,8)p_2 + (3,8)p_3 + (4,8)p_4 + (5,8)p_5 + (6,8)p_6 + (7,8)p_7 - (8,9)p_8 \\ p_9' = (3,9)p_3 + (4,9)p_4 + (6,9)p_6 + (8,9)p_8 \end{cases} \qquad (10)$$

Replacing the states' transition rates from table 4 in formula (9), it results the system:

$$\begin{cases} p_0^{'} = -(\lambda_{SW} * C_{dSW} + \lambda_{SW} * C_{dSW} * C_{iSW} * C_{rSW})p_0 - \lambda_{SW}(1 - C_{dSW})p_0 - \\ \quad - (\lambda_{HW} * C_{dHW} + \lambda_{HW} * C_{dHW} * C_{iHW} * C_{rHW})p_0 - \lambda_{HW}(1 - C_{dHW})p_0 - \\ \quad - \lambda_{SW}(1 - C_{dSW} * C_{iSW} * C_{rSW})p_0 - \lambda_{HW}(1 - C_{dHW} * C_{iHW} * C_{rHW})p_0 + \mu_{SW}p_1 + \mu_{HW}p_3 \\ p_1^{'} = (\lambda_{SW} * C_{dSW} + \lambda_S * C_{dSW} * C_{iSW} * C_{rSW})p_0 - [\mu_{SW} + \lambda_{HW} + \lambda_{SW} + \lambda_{HW}]p_1 + \\ \quad + (Tr_{SW})^{-1} p_5 + \mu_{SW}p_7 + \mu_{HW}p_8 \\ p_2^{'} = \lambda_{SW}(1 - C_{dSW})p_0 - [\lambda_{HW}(1 - C_{dHW}) + \lambda_{HW} * C_{dHW} + \lambda_{SW}]p_2 \\ p_3^{'} = (\lambda_{HW} * C_{dHW} + \lambda_H * C_{dHW} * C_{iHW} * C_{rHW})p_0 + \lambda_{HW}p_1 + (\lambda_{HW} * C_{dHW})p_2 - \\ \quad - [\mu_{HW} + \lambda_{SW} + \lambda_{HW}]p_3 + (\lambda_{HW} * C_{dHW})p_4 + \lambda_{HW}p_5 + (Tr_{HW})^{-1} p_6 + \mu_{SW}p_8 + \mu_{HW}p_9 \\ p_4^{'} = \lambda_{HW}(1 - C_{dHW})p_0 + \lambda_{HW}(1 - C_{dHW})p_2 - [\lambda_{HW} * C_{dHW} + \lambda_{SW} + \lambda_{HW}]p_4 \\ p_5^{'} = \lambda_{SW}(1 - C_{dSW} * C_{iSW} * C_{rSW})p_0 - [(Tr_{SW})^{-1} + \lambda_{HW} + \lambda_{SW} + \lambda_{HW}]p_5 \\ p_6^{'} = \lambda_{HW}(1 - C_{dHW} * C_{iHW} * C_{rHW})p_0 - [(Tr_{HW})^{-1} + \lambda_{SW} + \lambda_{HW}]p_6 \\ p_7^{'} = \lambda_{SW}p_1 + \lambda_{SW}p_2 + \lambda_{SW}p_5 - [\mu_{SW} + 2\lambda_{HW}]p_7 \\ p_8^{'} = \lambda_{HW}p_1 + \lambda_{HW}p_2 + \lambda_{SW}p_3 + \lambda_{SW}p_4 + \lambda_{HW}p_5 + \lambda_{SW}p_6 + 2\lambda_{HW}p_7 - [\mu_{HW} + \mu_{SW} + \lambda_{HW}]p_8 \\ p_9^{'} = \lambda_{HW}p_3 + \lambda_{HW}p_4 + \lambda_{HW}p_6 + \lambda_{HW}p_8 - \mu_{HW}p_9 \end{cases} \tag{11}$$

that could be represented in canonical format as:

$$\begin{cases} p_0^{'} = -2(\lambda_{SW} + \lambda_{HW})p_0 + \mu_{SW}p_1 + \mu_{HW}p_3 \\ p_1^{'} = (\lambda_{SW} * C_{dSW} + \lambda_S * C_{dSW} * C_{iSW} * C_{rSW})p_0 - [\mu_{SW} + \lambda_{HW} + \lambda_{SW} + \lambda_{HW}]p_1 + (Tr_{SW})^{-1} p_5 + \mu_{SW}p_7 + \mu_{HW}p_8 \\ p_2^{'} = \lambda_{SW}(1 - C_{dSW})p_0 - [\lambda_{HW}(1 - C_{dHW}) + \lambda_{HW} * C_{dHW} + \lambda_{SW}]p_2 \\ p_3^{'} = (\lambda_{HW} * C_{dHW} + \lambda_H * C_{dHW} * C_{iHW} * C_{rHW})p_0 + \lambda_{HW}p_1 + (\lambda_{HW} * C_{dHW})p_2 - \\ \quad - [\mu_{HW} + \lambda_{SW} + \lambda_{HW}]p_3 + (\lambda_{HW} * C_{dHW})p_4 + \lambda_{HW}p_5 + (Tr_{HW})^{-1} p_6 + \mu_{SW}p_8 + \mu_{HW}p_9 \\ p_4^{'} = \lambda_{HW}(1 - C_{dHW})p_0 + \lambda_{HW}(1 - C_{dHW})p_2 - [\lambda_{HW} * C_{dHW} + \lambda_{SW} + \lambda_{HW}]p_4 \\ p_5^{'} = \lambda_{SW}(1 - C_{dSW} * C_{iSW} * C_{rSW})p_0 - [(Tr_{SW})^{-1} + \lambda_{HW} + \lambda_{SW} + \lambda_{HW}]p_5 \\ p_6^{'} = \lambda_{HW}(1 - C_{dHW} * C_{iHW} * C_{rHW})p_0 - [(Tr_{HW})^{-1} + \lambda_{SW} + \lambda_{HW}]p_6 \\ p_7^{'} = \lambda_{SW}p_1 + \lambda_{SW}p_2 + \lambda_{SW}p_5 - [\mu_{SW} + 2\lambda_{HW}]p_7 \\ p_8^{'} = \lambda_{HW}p_1 + \lambda_{HW}p_2 + \lambda_{SW}p_3 + \lambda_{SW}p_4 + \lambda_{HW}p_5 + \lambda_{SW}p_6 + 2\lambda_{HW}p_7 - [\mu_{HW} + \mu_{SW} + \lambda_{HW}]p_8 \\ p_9^{'} = \lambda_{HW}p_3 + \lambda_{HW}p_4 + \lambda_{HW}p_6 + \lambda_{HW}p_8 - \mu_{HW}p_9 \end{cases} \tag{12}$$

Also, replacing the states' transition rates from table 4 in formula (10), it results:

$$\begin{cases} p_0^{'} = -(\lambda_{SW} * C_{dSW} + \lambda_{SW} * C_{dSW} * C_{iSW} * C_{rSW})p_0 - \lambda_{SW}(1 - C_{dSW})p_0 - \\ \quad - (\lambda_{HW} * C_{dHW} + \lambda_{HW} * C_{dHW} * C_{iHW} * C_{rHW})p_0 - \lambda_{HW}(1 - C_{dHW})p_0 - \\ \quad - \lambda_{SW}(1 - C_{dSW} * C_{iSW} * C_{rSW})p_0 - \lambda_{HW}(1 - C_{dHW} * C_{iHW} * C_{rHW})p_0 + \mu_{SW}p_1 + \mu_{HW}p_3 \\ p_1^{'} = (\lambda_{SW} * C_{dSW} + \lambda_{SW} * C_{dSW} * C_{iSW} * C_{rSW})p_0 - [\mu_{SW} + \lambda_{HW} + \lambda_{SW} + \lambda_{HW}]p_1 \\ p_2^{'} = \lambda_{SW}(1 - C_{dSW})p_0 - [\lambda_{HW}(1 - C_{dHW}) + \lambda_{HW} * C_{dHW} + \lambda_{SW}]p_2 \\ p_3^{'} = (\lambda_{HW} * C_{dHW} + \lambda_{HW} * C_{dHW} * C_{iHW} * C_{rHW})p_0 + \lambda_{HW}p_1 + (\lambda_{HW} * C_{dHW})p_2 - \\ \quad - [\mu_{HW} + \lambda_{SW} + \lambda_{HW}]p_3 + (\lambda_{HW} * C_{dHW})p_4 \\ p_4^{'} = \lambda_{HW}(1 - C_{dHW})p_0 + \lambda_{HW}(1 - C_{dHW})p_2 - [\lambda_{HW} * C_{dHW} + \lambda_{SW} + \lambda_{HW}]p_4 \\ p_5^{'} = \lambda_{SW}(1 - C_{dSW} * C_{iSW} * C_{rSW})p_0 - [\lambda_{SW} + \lambda_{HW}]p_5 \\ p_6^{'} = \lambda_{HW}(1 - C_{dHW} * C_{iHW} * C_{rHW})p_0 - [\lambda_{SW} + \lambda_{HW}]p_6 \\ p_7^{'} = \lambda_{SW}p_1 + \lambda_{SW}p_2 + \lambda_{SW}p_5 - 2\lambda_{HW}p_7 \\ p_8^{'} = \lambda_{HW}p_1 + \lambda_{HW}p_2 + \lambda_{SW}p_3 + \lambda_{SW}p_4 + \lambda_{HW}p_5 + \lambda_{SW}p_6 + 2\lambda_{HW}p_7 - \lambda_{HW}p_8 \\ p_9^{'} = \lambda_{HW}p_3 + \lambda_{HW}p_4 + \lambda_{HW}p_6 + \lambda_{HW}p_8 \end{cases} \tag{13}$$

The system can be reduced to:

$$\begin{cases} p_0^{'} = -2(\lambda_{SW} + \lambda_{HW})p_0 + \mu_{SW}p_1 + \mu_{HW}p_3 \\ p_1^{'} = (\lambda_{SW} * C_{dSW} + \lambda_{SW} * C_{dSW} * C_{iSW} * C_{rSW})p_0 - [\mu_{SW} + \lambda_{HW} + \lambda_{SW} + \lambda_{HW}]p_1 \\ p_2^{'} = \lambda_{SW}(1 - C_{dSW})p_0 - [\lambda_{HW}(1 - C_{dHW}) + \lambda_{HW} * C_{dHW} + \lambda_{SW}]p_2 \\ p_3^{'} = (\lambda_{HW} * C_{dHW} + \lambda_{HW} * C_{dHW} * C_{iHW} * C_{rHW})p_0 + \lambda_{HW}p_1 + (\lambda_{HW} * C_{dHW})p_2 - \\ \quad - [\mu_{HW} + \lambda_{SW} + \lambda_{HW}]p_3 + (\lambda_{HW} * C_{dHW})p_4 \\ p_4^{'} = \lambda_{HW}(1 - C_{dHW})p_0 + \lambda_{HW}(1 - C_{dHW})p_2 - [\lambda_{HW} * C_{dHW} + \lambda_{SW} + \lambda_{HW}]p_4 \\ p_5^{'} = \lambda_{SW}(1 - C_{dSW} * C_{iSW} * C_{rSW})p_0 - [\lambda_{SW} + \lambda_{HW}]p_5 \\ p_6^{'} = \lambda_{HW}(1 - C_{dHW} * C_{iHW} * C_{rHW})p_0 - [\lambda_{SW} + \lambda_{HW}]p_6 \\ p_7^{'} = \lambda_{SW}p_1 + \lambda_{SW}p_2 + \lambda_{SW}p_5 - 2\lambda_{HW}p_7 \\ p_8^{'} = \lambda_{HW}p_1 + \lambda_{HW}p_2 + \lambda_{SW}p_3 + \lambda_{SW}p_4 + \lambda_{HW}p_5 + \lambda_{SW}p_6 + 2\lambda_{HW}p_7 - \lambda_{HW}p_8 \\ p_9^{'} = \lambda_{HW}p_3 + \lambda_{HW}p_4 + \lambda_{HW}p_6 + \lambda_{HW}p_8 \end{cases} \tag{14}$$

A specification has to be made: to solve the equations systems (12) and (14) - which are differential equations systems in Cauchy canonical format - specialized software is required (e.g. MathLab or MathCad).

When data regarding software and hardware failure rates and probabilities C_{dSW}, C_{dHW}, C_{iSW}, C_{iHW}, C_{rSW}, C_{rHW} are known or it had been inferred analytically and/or statistically, the calculus could go further to determine the equations systems' solutions, for the initial condition $p_0(0) = 0;\ p_k(0) = 0;\ k > 1$.

Once calculated the differential equations systems' solutions, the reliability indicators of the system (availability (D) and working probability $P(t)$) are determined using the formulas from paragraph 2.

Also the following indicators are determined using the value of the system's failure probability:

$$F_{def(i)}(t) = 1 - P(t) \tag{15}$$

respectively

$$T_{def(i)} = \int_0^{\infty} P(t)dt\,. \tag{16}$$

3. CONCLUSIONS

The findings proposed in this paper give a methodological framework to the field of reliability modeling of C4ISR systems hardware/software components. In summary, the goal of this paper is to propose certain contribution, such as:

(1) *Elaboration of an improved Markov model regarding the reliability of C4ISR system's components* (paragraph 2) that augments the Markov model proposed by Rome Laboratory's methodology for hardware/software reliability modeling with information offered by general reliability theory in respect with Markov processes method. Synthesizing (based on the model) of new architectural products, specific to C4ISR system's Reliability Vision;

(2) *The implementation of the proposed Markov model for a representative case study* - modeling the reliability of a simple redundant hardware/software system, using a combination of system's states tables with states transition graphs and a matrix representation of fluency graphs.

BIBLIOGRAPHY

1. Ghiță, Al. & Ionescu, V. (1996). *Methods for Reliability Calculus.* Bucharest: Military Technical Academy Publishing House. 55-56.
2. Gupta, P. (2011). *Markov Modeling and Availability Analysis of a Chemical Production System-A Case Study*. Proceedings of the World Congress on Engineering, Vol I, July 2011, London, U.K.
3. Lai, C.D. *et al*. (2002). A model for availability analysis of distributed software/ hardware systems. *Information and Software Technology*, Vol 44, Issue 6. 346-347.
4. Lakey, P.B. & .Neufelder, A.M. (1997). *System and Software Reliability Assurance Notebook*. Rome Laboratory Report. Rome, NY: Griffins Air Force Base
5. Vasilescu, C. (2010). *The Reliability of Command and Control Systems within Romanian Air Force*. Brasov: Transilvania University Publishing House..