MODELING THE RELIABILITY OF INFORMATION MANAGEMENT SYSTEMS BASED ON MISSION SPECIFIC TOOLS SET SOFTWARE

Cezar VASILESCU

Associate Professor, Regional Department of Defense Resources Management Studies,
National Defense University, Bucharest, Romania

E-mail: caesarv@crmra.ro

Abstract: *The operational environments in which information management systems operate determine the existence of complex situations. Consequently, the command and control flow can take different paths, which involve different "sets" of activities. Each of those activities is associated with a specific software application set, known as Application Software Tools (ASTs).*
An operational profile represents a sequence of specific processing of distinct activities (from a functional point of view), based on specific Application Software Tools and with a certain time limit interval. Each operational profile has associated a probability of occurrence.
Each activity is performed during a specified period of time, with specific sets of ASTs. Totality resulting AST specification due to operational profiles crowd formed a mission specific software application system, also known as a Mission Specific Tools Set (MSTS). Each MSTS's element fulfill functions that meet the corresponding command and control activities, found in the form of lists of features of the system operational profile.
The aim of this paper is to present an original MSTS reliability model, which combines the modelling approach based on operational profiles with Rome Research Laboratory software reliability modeling methodology. In this way, it was realized a dual representation of application set's reliability that quantifies its level of reliability and also the associated weights of each application. The final goal was to offer an adequate basis for the process of reliability growth.
This paper is also going to provide a calculus example of MSTS system reliability using a representative U.S. Navy C4ISR system's combat action (Time Critical Targeting). The case study demonstrates the validity and the usefulness of the model in order to increase the system's reliability.

Key words: *Reliability modeling; Increase of software applications reliability; Operational profiles; Application software tools; Mission Specific Tools Set*

Introduction

Information management systems realize the processing of specific information necessary to conduct modern battlefield complex command and control activities, in order to ensure the success in battle. For mission-oriented software development it is necessary the modularization of the command and control activities and sub activities.

Generally, the operational profile can be defined as a quantitative characterization of the software usage, depending on the input space values. A profile consists of an independent possibilities set and their associated occurrence probabilities [6][1].

The operational environments in which information management systems operate determine the existence of complex situations, characterized by a great diversity of information, inputs, actualization operations etc. Consequently, the command and control flow can take different paths, which involve different "sets" of activities. Each of those activities is associated with a specific software application set, known as *Application Software Tools (ASTs)*.

Speaking about information management systems, an operational profile represents a sequence of specific processing of distinct activities (from a functional point of view), based on specific Application Software Tools and with a certain time limit interval. Each operational profile has associated a probability of occurrence.

Each activity is performed during a specified period of time, with specific sets of ASTs. Totality resulting AST specification due to operational profiles crowd formed a mission specific software application system, also known as a *Mission Specific Tools Set (MSTS)*.

Each MSTS's element fulfill functions that meet the corresponding command and control activities, found in the form of lists of features of the system operational profile.

Calculation of MSTS system reliability will be subject to of following paragraph.

Calculation of MSTS system reliability

MSTS system reliability prediction and growth requires a dual core computing. This approach is driven by the possibility of joint activities under different distinct operational profiles.

The calculation relations are:

$$R_{MSTS} = \sum_{k=1}^{N_P} p_k R_k \qquad (1)$$

in which

p_k - Occurrence probability of the k operational profile;

R_k - Reliability of the k operational profile;

N_P - Number of operational profiles.

The first relationship is based on the fact that each operational profile is associated with an occurrence probability [2].

Notation

$\alpha = \{\alpha_i; i = 1, N_\alpha\}$ = the set of MST activities;

$\alpha(k) = \{\alpha_i \in \alpha; \alpha_i \text{ belongs to the k profile}\}$, ranked in ascending order of execution in the profile;

$\varphi = \{AST : \text{AST is an instrument of the MSTS}\}$;

$\varphi(\alpha_i) = \{AST \in \varphi : \text{AST serves } \alpha_i\}$;

AST = specific software application sets.

Then

$$R_k = \prod_{\alpha_i \in \alpha(k)} R_{\alpha_i}\left(t''_{\alpha_i} - t'_{\alpha_i}\right) \quad (2)$$

in which

t'_{α_i} = the beginning moment of activity α_i;

t''_{α_i} = the ending moment of activity α_i.

and where

$$R_{\alpha_i}\left(t''_{\alpha_i} - t'_{\alpha_i}\right) = \prod_{AST \in \varphi(\alpha_i)} R_{AST}\left(t''_{\alpha_i} - t'_{\alpha_i}\right) \quad (3)$$

where

$$R_{AST}\left(t''_{\alpha_i} - t'_{\alpha_i}\right) = e^{-\lambda_{AST}\left(t''_{\alpha_i} - t'_{\alpha_i}\right)} \quad (4)$$

$$\lambda_{AST}\left(t''_{\alpha_i} - t'_{\alpha_i}\right) = \sum \lambda_{AST} \quad (4bis)$$

The second calculation relation of MSTS system reliability is:

$$R_{MSTS}^{dual} = \prod_{k=1}^{N_P} R_k \quad (1\ dual)$$

can be transformed as

$$R_{MSTS}^{dual} = \prod_{AST \in \varphi} R_{AST}^{*} \quad (5)$$

in which R_{AST}^{*} is the product of all factors in the formula (1 dual) that correspond to the same AST.

Note:

$R_{AST}^{*} = R_{AST}$ in case the AST appears one time in the formula (1 dual) and

$R_{AST}^{*} \neq R_{AST}$, $R_{AST}^{*} < R_{AST}$ otherwise.

This dualism is needed when profiles include joint activities. Using the first formula for calculating the reliability R_{MSTS} (in which components may occur several times) can provide the specification requirements for MSTS system reliability assessment and correspondence with the reliability requirements [1].

Also, MSTS system reliability calculation using the second relation (R_{MSTS}^{dual}) provides the possibility to organize the process of reliability growth, to meet the requirements specified. Thus in the calculation of reliability can be calculated weights (Π_{AST}) associated with AST and determined their influence.

$$\prod_{AST} = \ln R_{AST}^{*} / \ln R_{MSTS}^{dual} \quad (6)$$

followed by the increasing ordering of the resulting string of weights $\{\Pi_{AST} : AST \in \varphi\}$, to highlight the order of priorities in addressing the growth of MSTS reliability. In this way can be highlighted MSTS components unsatisfactory in terms of reliability, so giving a good support to system software designers to eventually redesign it (if required) in the process of reliability growth.

In what follows, we present an example of calculating the MSTS system reliability, for the most common case in practical operation of the information management systems, in which under different operational profiles are common joint activities.

Case study

Depending on the nature, size and membership of the information management systems to a category of forces or other, command and control activities can have a high degree of specificity. In [3] there have been listed a number of typical command and control activities, and the general categories from which they belong. Also, in case of large information management systems analysis (e.g. national level), identification and analysis of all activities can be difficult.

For this reason, we calculate the MSTS system reliability [4] using for example one of the U.S. Navy C4ISR system's combat action. For this, it is necessary a brief overview of the C4ISR system and command and control activities related to combat action "Time Critical Targeting" [5].

U.S. Navy uses various systems against naval and air targets, with different C4ISR systems providing guidance. The flow of activities involved was analyzed, in order to optimize command and control activities, eliminate the overlapping functionality and ensure interoperability of systems.

Table 1. The main command and control sub activities, related with the combat action "Time Critical Targeting"

Current issue	Name
1.1.	Analysis of surveillance and reconnaissance data list
2.1.	Reconcile target combat priorities
2.2.	Determine sensor availability
2.3.	Task sensor
2.4.	Collect data
3.1.	Detect target
3.2.	Determine environment
3.3.	Tracking and positioning the target

Current issue	Name
3.4.	Identifying target
4.1.	Update target list
4.2.	Assess engagement capability
4.3.	Assign weapon-target Platform selection
4.4.	Update mission plans
4.5.	Perform TCT (time critical target) deconfliction
5.1.	Execute force order
5.2.	Support weapon flyout
5.3.	Fighting target
6.1.	Collect information on damage
6.2.	Damage information assessment
6.3	Remove objective from target list

The flow of command and control activities related to combat action "Time Critical Targeting" (according to Table 1) is shown in Figure 1.

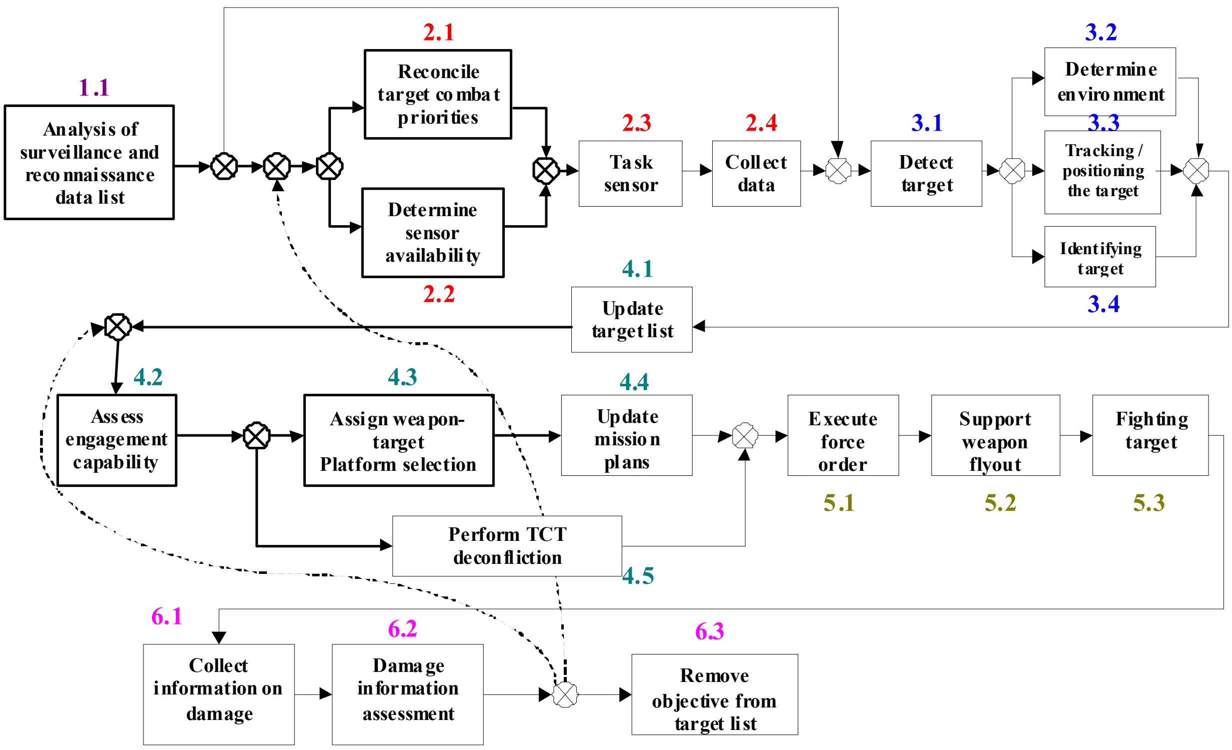

Figure 1. Scenario of C2 sub activities related to "Time Critical Targeting" combat action

AST names associated with sub activities are not relevant to the proposed goals. We present in terms of quantity the correlation between sub-C4ISR activities contained in Figure 1 and the number of software modules providing support to their deployment (Figure 2).

Typically, each operational profile of C4ISR activities is a chain of sequential actions. Application software tools sets are executed sequentially and/or competitive (exits a set representing the input for another set).

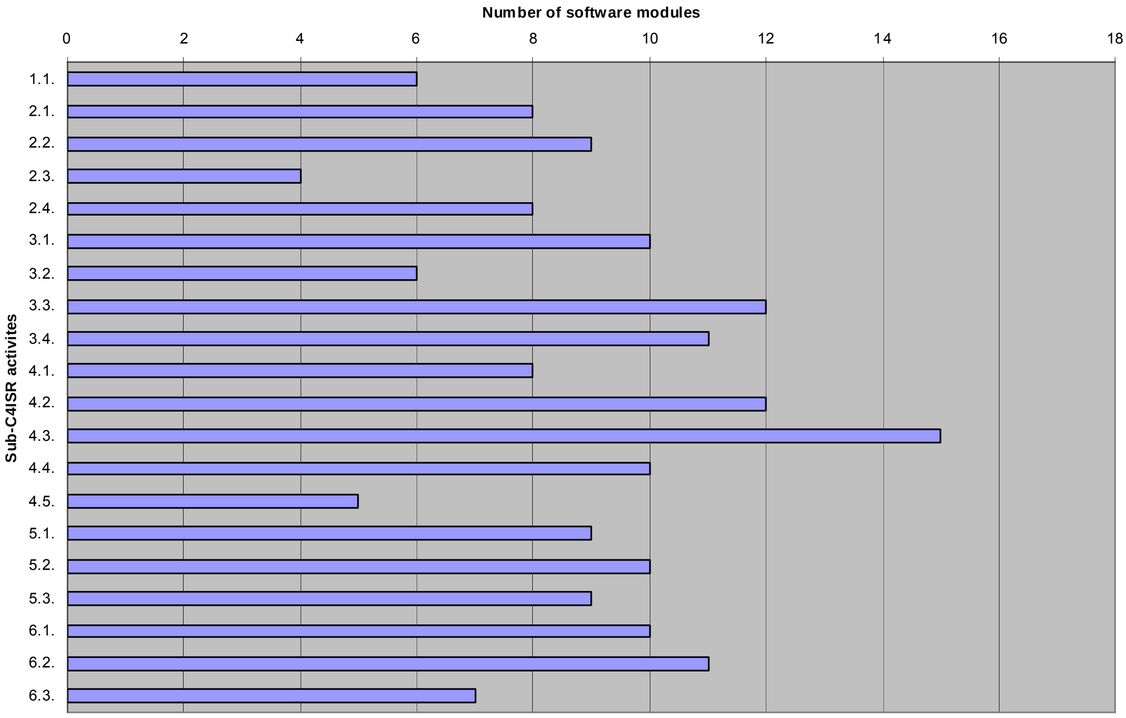

Figure 2. Graph of the number of software modules providing support to consisting sub-activities of "Time Critical Targeting" activity

We analyze the scenario of C4ISR sub activities related to "Time Critical Targeting" combat action (figure 1) to determine the operational profiles [4]. As a working hypothesis, we consider the entry of only one aircraft in the system (potential target) and use those numbers to each activity according to figure. The data related with operation of system's software modules (values estimated for failure rates by type of software modules and times of activation, ie completion) were altered to serve for illustration purposes.

Step 1

Determine operational profiles (sequences of activities):

– *profile 1* (target entry into the system, fight and destroy it)

$$(1.1)\to(2.1)\to(2.2)\to(2.3)\to(2.4)\to(3.1)\to(3.2)\to(3.3)\to(3.4)\to(4.1)\to(4.2)\to$$
$$\to(4.3)\to(4.4)\to(4.5)\to(5.1)\to(5.2)\to(5.3)\to(6.1)\to(6.2)\to(6.3)$$

– *profile 2* (target already challenged but still undamaged)

$$(4.2)\to(4.3)\to(4.4)\to(4.5)\to(5.1)\to(5.2)\to(5.3)\to(6.1)\to(6.2)\to(6.3)$$

– *profile 3* (target already challenged, still undamaged and emerged from the initial radar surveillance sector)

$$(2.1)\to(2.2)\to(2.3)\to(2.4)\to(3.1)\to(3.2)\to(3.3)\to(3.4)\to(4.1)\to(4.2)\to(4.3)\to$$
$$\to(4.4)\to(4.5)\to(5.1)\to(5.2)\to(5.3)\to(6.1)\to(6.2)\to(6.3)$$

Each C4ISR activity is done through a variable number of specific sets of software applications (AST). In turn, each AST consists of a variable number of independent software modules executed competitively (Table 2), whose characteristics are presented in Table 3.

Table 2. Correspondence between C4ISR activities, specific sets of software applications and number of software modules

C4ISR activities	Specific sets of software applications (AST)	Number of software modules
1	1.1.	6
2	2.1.	8
	2.2.	9
	2.3.	4
	2.4.	8
3	3.1.	10
	3.2.	6
	3.3.	12
	3.4.	11
4	4.1.	8
	4.2.	12
	4.3.	15
	4.4.	10
	4.5.	5
5	5.1.	9
	5.2.	10
	5.3.	9
6	6.1.	10
	6.2.	11
	6.3.	7

Step 2

We calculate for each AST the average failure rate and the reliability during operation.

We present detailed calculations for AST 1.1 and AST 2.1, following that for others to mention only the final results.

The average failure rate for AST is calculated using the equation:

$$\lambda_{AST} = \sum_{i=1}^{m} \lambda_{AST_i} \text{ ,}$$

where

m = number of competitive active software modules corresponding to AST

$$\lambda_{AST1.1} = \sum_{i=1}^{m} \lambda_{AST_i} = (3+6+2+8+4+8)\times 10^{-5} = 0{,}00031 \text{ hours}^{-1}$$

$$\lambda_{AST2.1} = \sum_{i=1}^{m} \lambda_{AST_i} = (3+6+3+7+8+7+8+3)\times 10^{-5} = 0{,}00045 \text{ hours}^{-1}$$

Reliability function will be:

$$R_{AST1.1} = e^{-\lambda_{AST1.1}} = e^{-0{,}00031} = 0{,}99969$$

$$R_{AST2.1} = e^{-\lambda_{AST2.1}} = e^{-0{,}00045} = 0{,}99955$$

Table 4 presents values of average failure rates and reliability of specific application software sets.

Table 3. Characteristics of software modules sequentially active

AST	/ Types of software modules	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.1	Activation time	0	45	200	300	800	900	-	-	-	-	-	-	-	-	-
	Completion time	45	200	300	800	900	1000	-	-	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	3	6	2	8	4	8	-	-	-	-	-	-	-	-	-
2.1	Activation time	0	50	100	250	400	600	750	980	-	-	-	-	-	-	-
	Completion time	50	100	250	400	600	750	980	1200	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	3	6	3	7	8	7	8	3	-	-	-	-	-	-	-
2.2	Activation time	0	55	100	150	300	500	650	800	1050	-	-	-	-	-	-
	Completion time	55	100	150	300	500	650	800	1050	1150	-	-	-	-	-	-
	Failure rate (x10^{-5})	5	1	2	5	7	4	8	3	7	-	-	-	-	-	-
2.3	Activation time	0	45	200	300	-	-	-	-	-	-	-	-	-	-	-
	Completion time	45	200	300	600	-	-	-	-	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	2	8	4	5	-	-	-	-	-	-	-	-	-	-	-
2.4	Activation time	0	70	160	250	450	600	900	1000	-	-	-	-	-	-	-
	Completion time	70	160	250	450	600	900	1000	1200	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	3	6	7	2	1	8	7	4	-	-	-	-	-	-	-
3.1	Activation time	0	115	200	300	500	700	1000	1150	1240	1350	-	-	-	-	-
	Completion time	115	200	300	500	700	1000	1150	1240	1350	1500	-	-	-	-	-
	Failure rate (x10^{-5})	1	2	8	5	5	3	6	8	8	9	-	-	-	-	-
3.2	Activation time	0	85	200	400	700	900	-	-	-	-	-	-	-	-	-
	Completion time	85	200	400	700	900	1000	-	-	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	2	4	4	2	5	7	-	-	-	-	-	-	-	-	-
3.3	Activation time	0	50	150	300	450	700	860	940	1025	1200	1350	1420	-	-	-
	Completion time	50	150	300	450	700	860	940	1025	1200	1350	1420	1550	-	-	-
	Failure rate (x10^{-5})	1	8	2	5	6	6	8	8	8	2	3	4	-	-	-
3.4	Activation time	0	85	150	250	500	600	850	930	1020	1250	1450	-	-	-	-
	Completion time	85	150	250	500	600	850	930	1020	1250	1450	1590	-	-	-	-
	Failure rate (x10^{-5})	2	2	3	7	5	6	1	8	5	8	7	-	-	-	-
4.1	Activation time	0	50	100	200	400	650	800	900	-	-	-	-	-	-	-
	Completion time	50	100	200	400	650	800	900	1050	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	3	5	6	5	3	4	2	7	-	-	-	-	-	-	-
4.2	Activation time	0	100	150	300	450	700	840	930	1000	1150	1320	1450	-	-	-
	Completion time	100	150	300	450	700	840	930	1000	1150	1320	1450	1600	-	-	-
	Failure rate (x10^{-5})	3	3	3	4	8	6	2	4	3	7	4	1	-	-	-
4.3	Activation time	0	75	130	250	400	500	740	820	1000	1090	1230	1310	1500	1600	1690
	Completion time	75	130	250	400	500	740	820	1000	1090	1230	1310	1500	1600	1690	1820
	Failure rate (x10^{-5})	2	4	3	5	8	2	3	7	6	6	6	4	3	2	9
4.4	Activation time	0	95	200	350	600	900	1100	1260	1450	1600	-	-	-	-	-
	Completion time	95	200	350	600	900	1100	1260	1450	1600	1800	-	-	-	-	-
	Failure rate (x10^{-5})	2	1	2	3	6	7	8	6	4	4	-	-	-	-	-
4.5	Activation time	0	65	200	500	800	-	-	-	-	-	-	-	-	-	-
	Completion time	65	200	500	800	900	-	-	-	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	3	3	5	6	2	-	-	-	-	-	-	-	-	-	-
5.1	Activation time	0	100	200	300	400	500	750	850	1000	-	-	-	-	-	-
	Completion time	100	200	300	400	500	750	850	1000	1200	-	-	-	-	-	-
	Failure rate (x10^{-5})	8	3	5	2	7	6	5	4	4	-	-	-	-	-	-
5.2	Activation time	0	115	200	350	800	900	1100	1300	1780	2000	-	-	-	-	-
	Completion time	115	200	350	800	900	1100	1300	1780	2000	2300	-	-	-	-	-
	Failure rate (x10^{-5})	1	3	3	6	8	9	2	6	6	4	-	-	-	-	-
5.3	Activation time	0	100	300	500	700	900	1000	1200	1290	-	-	-	-	-	-
	Completion time	100	300	500	700	900	1000	1200	1290	1500	-	-	-	-	-	-
	Failure rate (x10^{-5})	6	2	3	8	3	5	5	5	6	-	-	-	-	-	-
6.1	Activation time	0	55	120	450	670	800	895	975	1056	1170	-	-	-	-	-
	Completion time	55	120	450	670	800	895	975	1056	1170	1300	-	-	-	-	-
	Failure rate (x10^{-5})	2	3	6	3	1	2	6	7	8	3	-	-	-	-	-
6.2	Activation time	0	105	200	350	600	800	980	1100	1200	1350	1440	-	-	-	-
	Completion time	105	200	350	600	800	980	1100	1200	1350	1440	1565	-	-	-	-
	Failure rate (x10^{-5})	3	3	4	3	4	3	2	6	6	8	5	-	-	-	-

AST	/ Types of software modules	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6.3	Activation time	0	100	200	330	500	700	860	-	-	-	-	-	-	-	-
	Completion time	100	200	330	500	700	860	1200	-	-	-	-	-	-	-	-
	Failure rate (x10^{-5})	5	6	3	4	1	7	8	-	-	-	-	-	-	-	-

Table 4. Values of average failure rates and reliability of specific application software sets

AST	λ_{AST}	R_{AST}
1.1	0,00031	0,99969
2.1	0,00045	0,99955
2.2	0,00042	0,99958
2.3	0,00019	0,99981
2.4	0,00038	0,99962
3.1	0,00055	0,99945
3.2	0,00024	0,99976
3.3	0,00061	0,99939
3.4	0,00054	0,99946
4.1	0,00035	0,99965
4.2	0,00048	0,99952
4.3	0,00070	0,99930
4.4	0,00043	0,99957
4.5	0,00019	0,99981
5.1	0,00044	0,99956
5.2	0,00048	0,99952
5.3	0,00043	0,99957
6.1	0,00041	0,99959
6.2	0,00047	0,99953
6.3	0,00034	0,99966

Step 3

We calculate the reliability of C4ISR activities R_{α_i}.

$$R_{\alpha_i} = \prod_{AST \in \varphi(\alpha_i)} R_{AST}$$

$$R_{\alpha_1} = R_{AST1.1} = 0{,}99969$$

$$R_{\alpha_2} = R_{AST2.1} \times R_{AST2.2} \times R_{AST2.3} \times R_{AST2.4} = 0{,}99865$$

$$R_{\alpha_3} = R_{AST3.1} \times R_{AST3.2} \times R_{AST3.3} \times R_{AST3.4} = 0{,}99806$$

$$R_{\alpha_4} = R_{AST4.1} \times R_{AST4.2} \times R_{AST4.3} \times R_{AST4.4} \times R_{AST4.5} = 0{,}99785$$

$$R_{\alpha_5} = R_{AST5.1} \times R_{AST5.2} \times R_{AST5.3} = 0{,}99865$$

$$R_{\alpha_6} = R_{AST6.1} \times R_{AST6.2} \times R_{AST6.3} = 0{,}99878$$

Step 4

The reliability of operational profiles R_{pk} is:

$$R_1 = \prod R_{\alpha_i} = R_{\alpha_1} \times R_{\alpha_2} \times R_{\alpha_3} \times R_{\alpha_4} \times R_{\alpha_5} \times R_{\alpha_6} = 0{,}991625$$

$$R_2 = R_{\alpha_4} \times R_{\alpha_5} \times R_{\alpha_6} = 0{,}995291$$

$$R_3 = R_{\alpha_2} \times R_{\alpha_3} \times R_{\alpha_4} \times R_{\alpha_5} \times R_{\alpha_6} = 0{,}991933$$

Consider the following values for the operational profiles' probability of occurrence p_k :

$$p_1 = 0{,}75$$

$$p_2 = 0{,}15$$

$$p_3 = 0{,}10$$

Step 5

MSTS reliability is:

$$R_{MSTS} = \sum_{k=1}^{N_P} p_k R_k = \sum_{k=1}^{3} p_k R_k = p_1 R_1 + p_2 R_2 + p_3 R_3$$

$$R_{MSTS} = (0{,}75 \times 0{,}991625) + (0{,}15 \times 0{,}995291) + (0{,}10 \times 0{,}991933) = 0{,}992206$$

If using formula (1dual) for MSTS system's reliability calculation, we can rewrite step 5, as follows:

Step 5 (dual)

MSTS reliability is:

$$R_{MSTS}^{dual} = \prod_{k=1}^{N_P} R_k = \prod_{k=1}^{3} R_k = R_1 \times R_2 \times R_3 = 0{,}978994$$

Also, there is a new step:

Step 6

We calculate the weights $\prod_{AST}$ associated with each AST using the formula:

$$\prod_{AST} = \ln R_{AST}^{*} / \ln R_{MSTS}^{dual}$$

We present detailed calculations for AST 1.1 and AST 2.1 associated weights, following that for others to mention only the final results.

Table 5 present values of weights associated to each specific application software set.

$$\prod_{AST1.1} = \ln R_{AST1.1} / \ln R_{MSTS}^{dual} = \ln(0{,}99969) / \ln(0{,}978994) = 0{,}036510$$

$$\prod_{AST2.1} = \ln R_{AST2.1} / \ln R_{MSTS}^{dual} = \ln(0{,}99955) / \ln(0{,}978994) = 0{,}053003$$

$$\prod_{AST2.2} = \ln R_{AST2.2} / \ln R_{MSTS}^{dual} = \ln(0{,}99958) / \ln(0{,}978994) = 0{,}049470$$

Table 5. The values of weights associated with application software sets.

AST	R_{AST}	$\prod_{AST}$
1.1	0,99969	0,036510
2.1	0,99955	0,053003
2.2	0,99958	0,049470

AST	R_{AST}	$\prod_{AST}$
2.3	0,99981	0,022378
2.4	0,99962	0,044758
3.1	0,99945	0,064785
3.2	0,99976	0,028265
3.3	0,99939	0,071855
3.4	0,99946	0,063608
4.1	0,99965	0,041223
4.2	0,99952	0,056538
4.3	0,99930	0,082460
4.4	0,99957	0,050648
4.5	0,99981	0,022378
5.1	0,99956	0,051825
5.2	0,99952	0,056538
5.3	0,99957	0,050648
6.1	0,99959	0,048290
6.2	0,99953	0,055360
6.3	0,99966	0,040045

We execute the decreasing ordering of the weights result string.

$$\left(\Pi_{AST}\right)_{AST\in\varphi} = \left(\Pi_{4.3},\Pi_{3.3},\Pi_{3.1},\Pi_{3.4},\Pi_{4.2},\Pi_{5.2},\Pi_{6.2},\Pi_{2.1},\Pi_{5.1},\Pi_{4.4},\Pi_{5.3},\Pi_{2.2},\Pi_{6.1},\Pi_{2.4},\Pi_{4.1},\right.$$
$$\left.\Pi_{6.3},\Pi_{1.1},\Pi_{3.2},\Pi_{2.3},\Pi_{4.5}\right)$$

The conclusion offered by the decreasing ordering of this string is that AST4.3 and AST3.3 have the largest weight (influence) on the reliability of the whole, any redesign of the software modules that compose AST4.3 and AST3.3 being highly recommended in the reliability increasing process.

References

1. Ghita, A. and Ionescu, V. **Metode de calcul în fiabilitate**, Military Technical Academy Publishing House, Bucharest, 1996
2. Musa, J.D. **Operational Profiles in Software Reliability Engineering**, IEEE Software Magazine, March, 1993
3. Vasilescu, C. and Ghita, A. **Contributions to the field of C4ISR Systems Reliability Modeling**, „Proceedings of the XXXIst Annual Scientific Communications Session with international attendance", Military Technical Academy, Bucharest, 3-4 November 2005
4. Vasilescu, C. **Prezentare generală a sistemelor de comandă şi control**, National Scientific Communications Session, Air Force Academy, Brasov, November 2000
5. Vasilescu, C. **Probleme actuale în fiabilitatea sistemelor de comanda si control din Fortele Aeriene**, Editura Universitatii Transilvania din Brasov, 2009, pp. 236-245
6. * * * **DoD Architecture Framework version 1.0 Deskbook**, DoD Architecture Working Group, August 2003
7. * * * **System and Software Reliability Assurance Notebook**, produced for Rome Laboratory, New York, 1997

[1] Codification of references:

[1]	Ghita, A. and Ionescu, V. **Metode de calcul în fiabilitate**, Military Technical Academy Publishing House, Bucharest, 1996
[2]	Musa, J.D. **Operational Profiles in Software Reliability Engineering**, IEEE Software Magazine, March, 1993
[3]	Vasilescu, C. **Prezentare generală a sistemelor de comandă şi control**, National Scientific Communications Session, Air Force Academy, Brasov, November 2000
[4]	Vasilescu, C. and Ghita, A. **Contributions to the field of C4ISR Systems Reliability Modeling**, „Proceedings of the XXXIst Annual Scientific Communications Session with international attendance”, Military Technical Academy, Bucharest, 3-4 November 2005
[5]	* * * **DoD Architecture Framework version 1.0 Deskbook**, DoD Architecture Working Group, August 2003
[6]	* * * **System and Software Reliability Assurance Notebook**, produced for Rome Laboratory, New York, 1997
[7]	Vasilescu, C. **Probleme actuale în fiabilitatea sistemelor de comanda si control din Fortele Aeriene**, Editura Universitatii Transilvania din Brasov, 2009, pp. 236-245