computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

SECURITY AND AVAILABILITY MODELS FOR SMART BUILDING AUTOMATION SYSTEMS

Vyacheslav Kharchenko [1], Yuriy Ponochovnyi [2], Al-Sudani Mustafa Qahtan Abdulmunem [1], Artem Boyarchuk [1]

[1] Department of Computer Systems and Networks, National Aerospace University KhAI, Chkalova str., 17, 61070, Kharkov, Ukraine, V.Kharchenko@khai.edu, mostafahkahtan1@gmail.com, a.boyarchuk@mail.ru, http://csn.khai.edu
[2] Department of Computer Engineering, Poltava National Technical University named after Yurij Kondratyuk, Pershotravneva Avenue, 24, 36011, Poltava, Ukraine, pnch1@rambler.ru, http://fitts.pntu.edu.ua

Abstract: This article presents the information on control system of smart building, which is considered as a set of subsystems including a building automation system. The paper considers the three-level architecture of the building automation system components, including FPGA, communication and management levels. It is determined that the causes of failures and inaccessibility of the BAS architecture component can be both internal system and external factors, among which software defects and vulnerabilities are identified. BAS security and availability during its life cycle are assessed using the Fault-, Attack- and Availability-Tree and Markov models. Markov model is used to develop a number of strategies which help to recover system and to eliminate all the possible threats during systems life time. The models of BAS architecture with software defects and attacked vulnerabilities with general reliability (defect) and security (vulnerability) maintenance are analyzed in detail. The recommendations on the choice of strategies and service parameters are given.

Keywords: Markov model, building automation system, smart home, maintenances strategies, common maintenance.

1. INTRODUCTION

The development of virtualization technology and the creation of cloud computing environments led to appearance of new variants of the IT systems architectures which need to be considered when assessing and ensuring the quality of modern computer systems and services known as a "smart home". This dynamic character of the processes of information interaction significantly complicates the possibility of rapid assessment of the reliability and availability of software and infrastructure resources available to remote access [1, 2].

Analysis of the system is performed in order to determine its dependability considering reliability and security issues; the method and techniques are developed in order to analyze the system availability, taking into account the value of parameter and how it can affect availability [16]. As the first step of analysis the Availability Tree Analysis (AvTA) has been previously used [3, 4, 9]; it helps to give wide picture for availability modeling. It is also aimed on simultaneous analysis of the attacks and faults during system life cycle. In addition, in this work, a number of strategies, which are used to analyze complex and big systems, have been developed. The architecture of these strategies depends on separate and common maintenance, which are analyzed in Sections 3 and 4.

According to the international standards [1, 5, 6], it is possible to assess the level of risk for a building an automation system and give requirements that need to be met in order to achieve the desired goal of safety and availability.

The primary objectives of the works [8, 11] are security issues for system design and the integration of security subsystems, which significantly tightens security requirements to the protocol of a network control system, and weaknesses in the system design according to hardware and software components. In [4, 20] it deals with development and research of Markov models of smart building automation systems (BAS), BAS failures can be caused by intra (reliability) reasons and external (security) reasons including software faults and attacks on vulnerabilities. The sets of faults and vulnerabilities are considered as separated ones, Markov models of BAS architecture with occurred software faults and attacked vulnerabilities considering three maintenance strategies are systematized and

developed. These strategies are based on recovery process without maintenance, maintenance with common and separate activities on reliability (faults) and vulnerabilities (security).

The typical definitions of 'defect' and 'failure' are used in this paper. Fault (defect) is an abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function. Failure is a termination of the ability of a functional unit to provide a required function or operation of a functional unit in any way other than as required. Error is a discrepancy between a computed, observed, or measured value, or condition and the true, specified or theoretically correct value or condition [21].

Recommendations concerning selection of strategies and parameters of maintenance are suggested.

In this paper, there will be a possibility to describe the main components that have an effect on the system availability, and help to minimize the time of analysis. In the second part of the work, a number of strategies using a Markov model are developed. These strategies deal with the system availability; it describes the possibility to recover from the down state (the state when there is a need to use these strategies) to the up state (the level of availability according to the customer requirements). The architecture of these strategies depends on the kind of maintenance (common or separate). The result of these strategies gives different ways for system recovering considering customer requirements as the maximum value of availability.

2. APPROACH AND MODELING TECHNIQUE

According to [11, 13, 15] BAS design has been divided into three levels and system availability depends on these levels; the components for each level have been analyzed by applying Attack Tree Analysis (ATA) [3, 9], which is used for security issues; and also a system fault has been analyzed using Fault Tree Analysis (FTA) [10, 12], which is implemented in order to analyze the failure issues. Fig. 1 shows the classification of BAS system analysis using combined AvTA method.

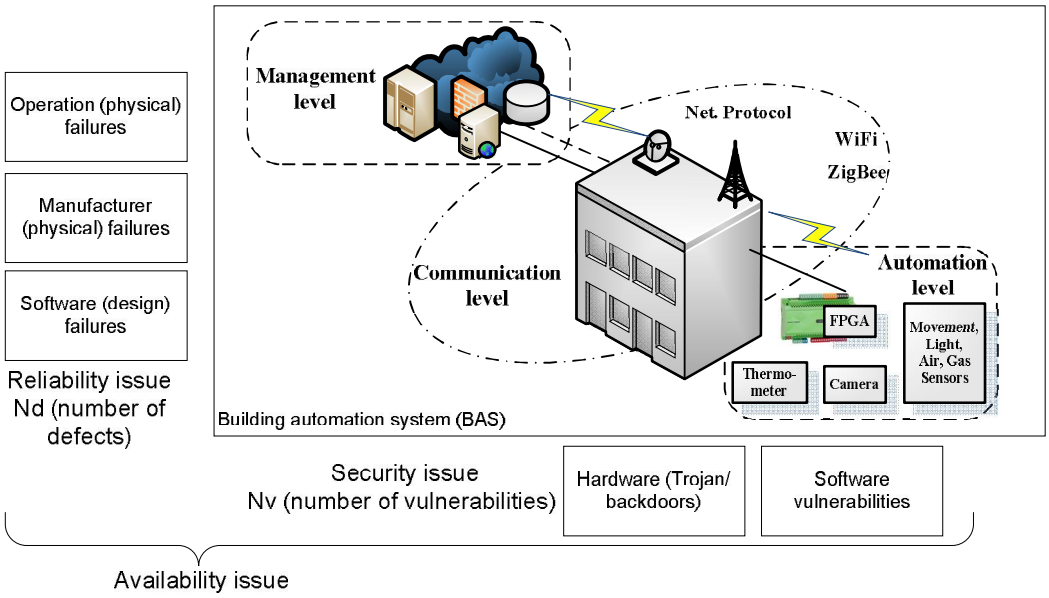

Fig. 1 – Availability analysis of building automation system with parameterization (Nd, Nv) of Markov model.

2.1 BUILDING AUTOMATION SYSTEM ARCHITECTURE AND COMPONENTS

BAS components are different items depending on the area of system application, but in general they can be separated as described in [7, 18]:

1. Upper level (Management Level): dispatching and administration functions, operations with databases and statistical functions. At this level the cooperation between personnel (operators, dispatchers etc.) and system is performed, which is implemented by computer devices and SCADA-systems. In our case study, the database of this level is analyzed considering reliability and security [17].

2. Middle level (Communication Level): responsible for communications between levels, and information sending/receiving. According to our analysis, Wireless Network has been chosen as one of these level components [15].

3. Low level (Automation Level): terminal level with input/output functions. This level includes sensors, actuating mechanisms, cabling between

devices and low-middle levels. FPGA is one of the important components used for this level [19].

These levels are divided depending on our vision system analysis. There are different designs of BAS but this design has been chosen as the easiest one for usage and analysis.

2.2 GENERAL MODELS (GRAPH)

We performed analysis of the BAS design in accordance to three levels, and made selection of the important components in system design (Wireless Network, FPGA, Database).

In this section the system will be analyzed considering described these parameters. If the analysis starts using the ATA methods, the result will be related to the security issues with the eliminated reliability as shown in Fig. 1.

The same situation will take place under implementation of the FTA. The analysis will be connected to reliability. In this case the AvTA method has been used, which allows to analyze system considering security and reliability issues at the same time. Also it shows the possibility of recovery of the components.

Our analysis deals with static and non-complex system; however, in case of a big system with different number of components it will be possible to use this method as well. In this step, a Markov model is developed. In the Markov model [3, 20] there is a possibility to add more components and eliminate them without any effect on the analysis process. During the first step in analysis process it is necessary to give a big picture for system with all possible states (Fig. 2).

The Markov model, which analyzes all possible states of the system, shows the transmission between operating states and recovery. Table 1 describes transmission and recovery between the states.

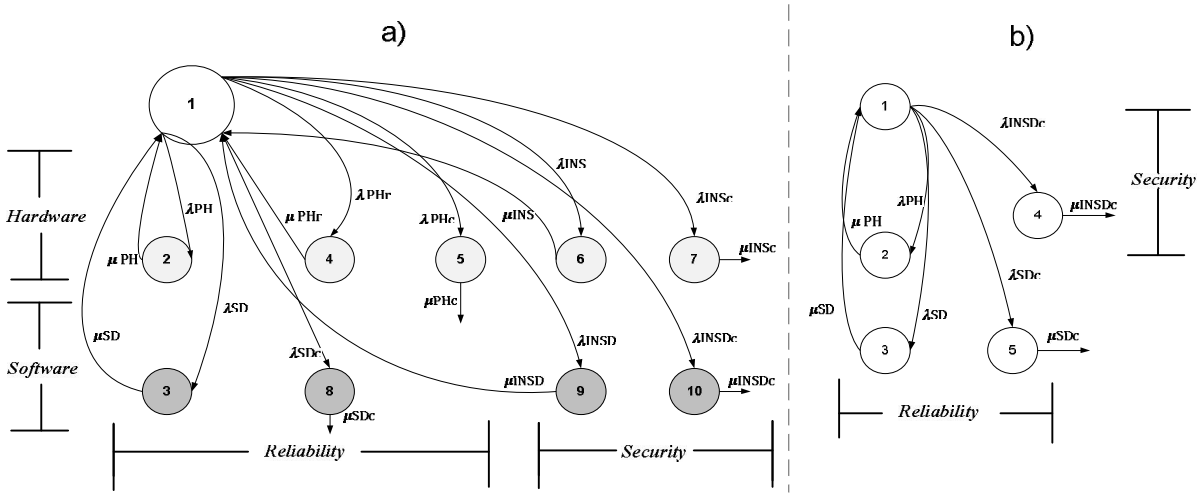

Fig. 2 – General (a) and Simplified (b) Markov graph of BAS availability.

Table 1. Description of parameters general models.

Param.	Failure/recovery rates	Param.	Failure/recovery rates
λPH	Physical operation failure (hardware)	μINSc	Intrusion failure (severe hardware vuln.erability/changing design)
μPH	Physical operation failure (hardware/repair)	λSD	Failure caused by design fault (software)
λPHr	Physical failure operation (soft error)	μSD	Soft error caused by design fault (software/restart)
μPHr	Phys. operation failure (soft hardware error/restart)	λSDc	Failure caused by design fault (software)
λPHc	Physical manufacture failure (hardware)	μSDc	Failure caused by design fault (soft. /changing code)
μPHc	Manufacture failure (hardware/changing design)	λINSD	Intrusion failure (soft software vulnerability)
λINS	Intrusion failure (soft hardware vulnerability)	μINSD	Intrusion failure (soft software vulnerability/restart)
μINS	Intrusion failure (soft hardware vuln. /restart)	λINSDc	Intrusion failure (severe software vulnerability)
λINSc	Intrusion failure (severe hardware vulnerability)	μINSDc	Intrusion failure (severe soft. vuln. /changing code)

2.3 MODEL SPECIFICATION

In this work, five models have been developed as shown in Table 2. The BAS analysis is divided into security issues part and reliability issues part. The states of transmission for Markov model is divided according to these two issues. First, the security part is presented as Nv (number of vulnerability); second is the reliability Nd (number of defects). The goal of these models is to eliminate Nv, Nd within minimum time of life cycle, and recover system to the maximum value of availability (A_{MBAS}constant) during period (T_{MBAS}constant).

Table 2. BAS models specification.

Type of service	Model name	Number of defects	Number of vulnerabilities	Number of services
-	MBAS1	0…Nd	0…Nv	0
common	MBAS2.1	0…Nd	0…Nv	∞
	MBAS2.2	0…Nd	0…Nv	0…Np
separate	MBAS3.1	0…Nd	0…Nv	∞
	MBAS3.2	0…Nd	0…Nv	0…Ndp, 0…Nvp

In some cases, the elimination process inside the system will not be able to eliminate the vulnerability or design fault; in this case, the maintenance strategies have been added, which support system to increase the elimination process. In our case, two types of maintenance strategies are used:

1. The common maintenance which deals with design fault and vulnerability in same time: the process of elimination will be sequential between design fault and vulnerability;

2. The separated maintenance, which deals with vulnerability and design fault separately one by one.

In next section, we describe the characteristics of maintenance strategies for two models: one with common maintenance and another with separated maintenance.

3. MARKOV MODEL FOR A LIMITED NUMBER OF COMMON MAINTENANCE

The Fig. 3 shows graph model with limited number of common maintenance. It is assumed that detection and removal of a software defect or vulnerability in the common maintenance is possible. A number of Np maintenance procedures are planned in the model.

The number of maintenance procedures Np is planned on the basis of design phase of BAS on the basis of expert evaluation.

Marked graph (Fig. 3) shows the BAS architecture with two defects and two vulnerabilities (Nd = 2, Nv = 2), in which six (Np = 6) common maintenance operations are performed.

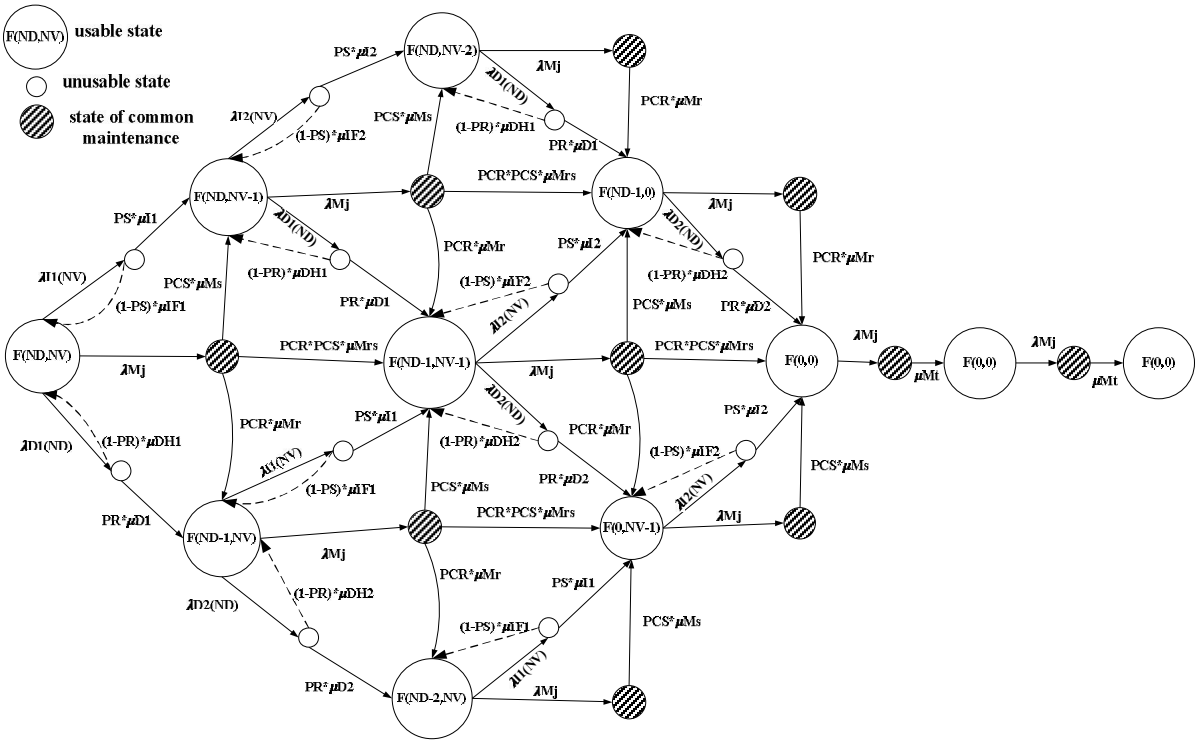

Fig. 3 – Marked graphs MBAS2.2 model taking into account common maintenance.

The parameter Np corresponds to the number of vertical diagonals of the rhomboid figure of oriented graph (where states of common maintenance are located). Such approach allows to confirm the condition of a Markov model. The overall number of potential states of common maintenance (10 potential states) is greater than actual number of maintenance states (Np = 6).

The logic of the model functioning in this case is the following: the first maintenance (Np = 1) is performed after system launch and its state has one input – transition from the state F(Nd, Nv). Further, different paths of movement over the states of model are possible, so the second maintenance (Np = 2) has

two probable states and is carried out either after elimination of defect (transition from state F(Nd-1,Nv)), or after removal of vulnerability (transition from state F(Nd, Nv-1)) or can be skipped (if during the first maintenance both fault and vulnerability are eliminated).

The third maintenance (Np=3) has three possible states (with transitions from states F(Nd,Nv-2), F(Nd-1,Nv-1), F(Nd-2,Nv)) and also can be skipped in case of identification and elimination defect and vulnerability during the second maintenance. The fourth maintenance (Np=4) has two possible states (with transitions from states F(Nd-1,0), F(0,Nv-1)); the fifth and sixth maintenances have one possible state (with the transition from the state F(0,0)).

Conditions of modeling the general maintenance procedure are shown by the shaded circles. Transitions in the service of the state carried out a usable state with the intensity of service λMj. During the event the software defect detection occurs with a probability PCR, detection of vulnerabilities - with probability PCS. Simultaneous detection of vulnerabilities and software defects occur with PCR*PCS probability. Previous events complements to a global group:

$$PCS + PCR + PCS * PCR = 1. \quad (1)$$

Thus, three transitions are possible:

a) in case of detection of vulnerabilities with probability PCS you can skip to the arrow straight up the intensity weighted PCS*μMs, where μMs - the inverse of the mean time to identify and eliminate vulnerabilities, μMs=1/(TdetV + TremV);

b) in the case of software defects with PCR likely to skip to a vertically downward direction, the weighted intensity of PCR*μMr, where μMr - the inverse of the mean time to identify and remedy the defect, μMr = 1/(TdetD + TremD);

c) in the case of a software defect detection and vulnerability likely PCS*PCR performed shift of the arrow to the right, the weighted intensity PCS*PCR*μMrs, where μMrs - the inverse of the average time detection and removal of defects and vulnerabilities, μMrs = μMr*μMs/(μMr + μMs).

Fig. 5 shows the graph of the MBAS2.2 model, in which the number of maintenances (Np=6) exceeds the actual number of system diagonals (Nd+Nv=4). So after removal of all defects and vulnerabilities, the common maintenance procedures are carried out during two more periods, and then they stop. In this regard, the availability function covers additional states and is calculated as follows:

$$A(t) = \sum_{i=0}^{Nk} P_i(t); \quad (2)$$
$$Nk = (Nd+1)*(Nv+1) + Np - (Nd+Nv) - 1$$

4. SIMULATION AND COMPARATIVE ANALYSIS

The following values of parameters for simulation were chosen (Table 3).

Results of simulation are illustrated by Fig. 4. The following conclusions can be formulated according to simulation of BAS with two different strategy of maintenance [2]. The analysis of the graphs in Fig. 4 showed that the limitation of the number of maintenances in MBAS2.2 and MBAS3.2 models makes it possible to achieve an ideal availability ($A_{MBAS2.2}const = A_{MBAS3.2}const = 1$) in the stable (stationary) mode. The minimum of availability function for models with limited and unlimited maintenance varies:

- 0.0057 with common maintenance (MBAS2.1 and MBAS2.2);
- 0.0161 with separate maintenance (MBAS3.1 and MBAS3.2).

The transition period for the stable mode for MBAS2.2 model is 2.5241 times higher than period for the MBAS3.2 separate maintenance model. At the same time, the elimination of defects and vulnerabilities in models with maintenance is faster than in the MBAS1 model (at least 3.7165 times).

As interest is caused by a decrease of period of detection and elimination of all defects and vulnerabilities, the influence of individual input parameters on the resulting indicator $T_{MBAS2.2}const$ is considered (in addition, their influence on $A_{MBAS2.2}min$ is analyzed). The dimensionality of the model is increased to Nd = 3, Nv = 3. The Np parameter varies from 0 to 10.

Table 3. Values of input parameters of simulation processing.

Symbol	Illustration	value	unit
laR(1)	The intensity of the first fault manifestation BAS λD1	5e-4	1/hour
laR(2)	The intensity of the second fault manifestation BAS λD2	4.5e-4	1/ hour
laS(1)	Intensity of the first vulnerability manifestation BAS λI1	3e-3	1/ hour
laS(2)	The intensity of the second vulnerability BAS λI2	3.5e-3	1/ hour
muR(1)	The intensity of the restoration with the removal of the first fault BAS μD1	0.5	1/ hour
muR(2)	The recovery rate with the elimination of the second fault BAS μD1	0.4	1/ hour
muS(1)	The recovery rate with the removal of the first vulnerability BAS μI1	0.45	1/ hour
muS(2)	The recovery rate with the elimination of the second vulnerability BAS μI2	0.34	1/ hour
muRH	The intensity of the restart without removing faults μDH1=μDH2	5	1/ hour

muSF	The intensity of the restart without removing vulnerability μIF1=μIF2	6	1/ hour
PR	The probability of fault elimination of the BAS during recovery	0.9	
PS	The probability of eliminating the vulnerability of the BAS during recovery	0.9	
laMj	The intensity of the common maintenance λMj	5e-3	1/hour
laMs	The intensity of the maintenance separate in vulnerabilities λMs	1e-3	1/hour
laMr	The intensity of the maintenance separate in defects λMr	2e-3	1/hour
muMt	The intensity of holding measures on common maintenance μMt	0.4	1/hour
muMs	The intensity of detecting and removing a vulnerability μMs	0.2	1/hour
muMr	The intensity of detecting and removing a defect μMr	0.3	1/hour
PCS	The probability of identifying vulnerabilities in the maintenance process	0.4409	
PCR	The probability of identifying a software defect in the maintenance process	0.388	

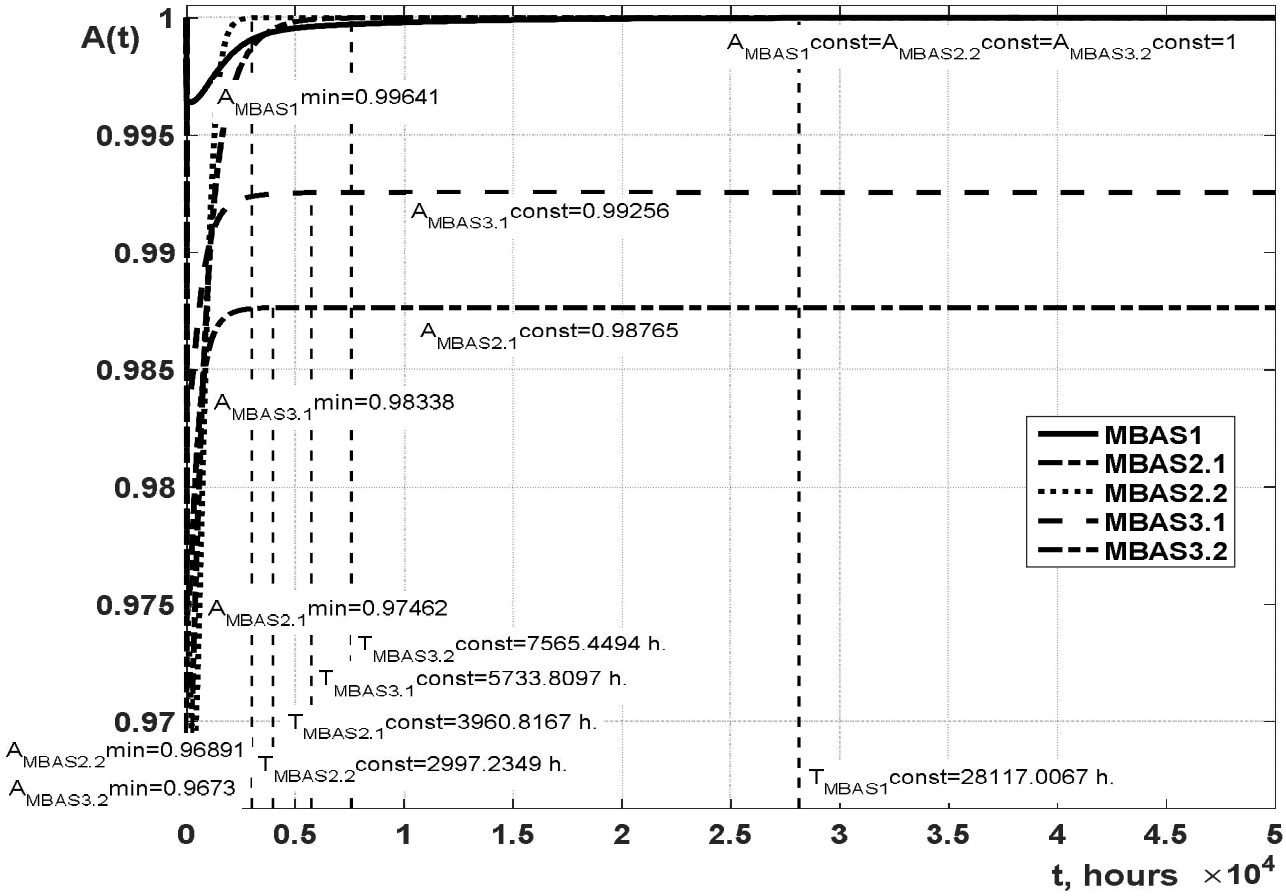

Fig. 4 – Simulation results of BAS architecture availability (the resulting figures are determined with an accuracy of 10^{-5}).

Research results of the effect of forecast accuracy (Np) have proved the expected result. If the lack of defects and vulnerabilities is predicted (Np = 0), the MBAS2.2 model degenerates into MBAS1 model (Fig. 5,a) and has the highest level of $A_{MBAS2.2}$min value (Fig. 5,c). With the growth of number of limited Np maintenances up to Np = 6, the process of identifying and eliminating defects and vulnerabilities is accelerating.

The graph of change of $T_{MBAS2.2}$const value in Fig. 5, d has a characteristic view of broken curve: up to the limit Np≤Nv + Nd, it shows a decrease of the resulting index, and for Np> Nv + Nd, the value of $T_{MBAS2.2}$const increases with Np growth (since non-result maintenance procedures accumulate).

A noticeable change in behavior of $A_{MBAS2.2}$min(Np) at Np = 5 is explained by the fact that with such a large number of maintenances the access is ensured from the service state of the extreme right operable state (Fig. 5,c). In this case, Fig. 5,a shows that with the appearance of excessive maintenances (Np = 6, Np = 8), the minimum of availability function shifts along the time axis to the right.

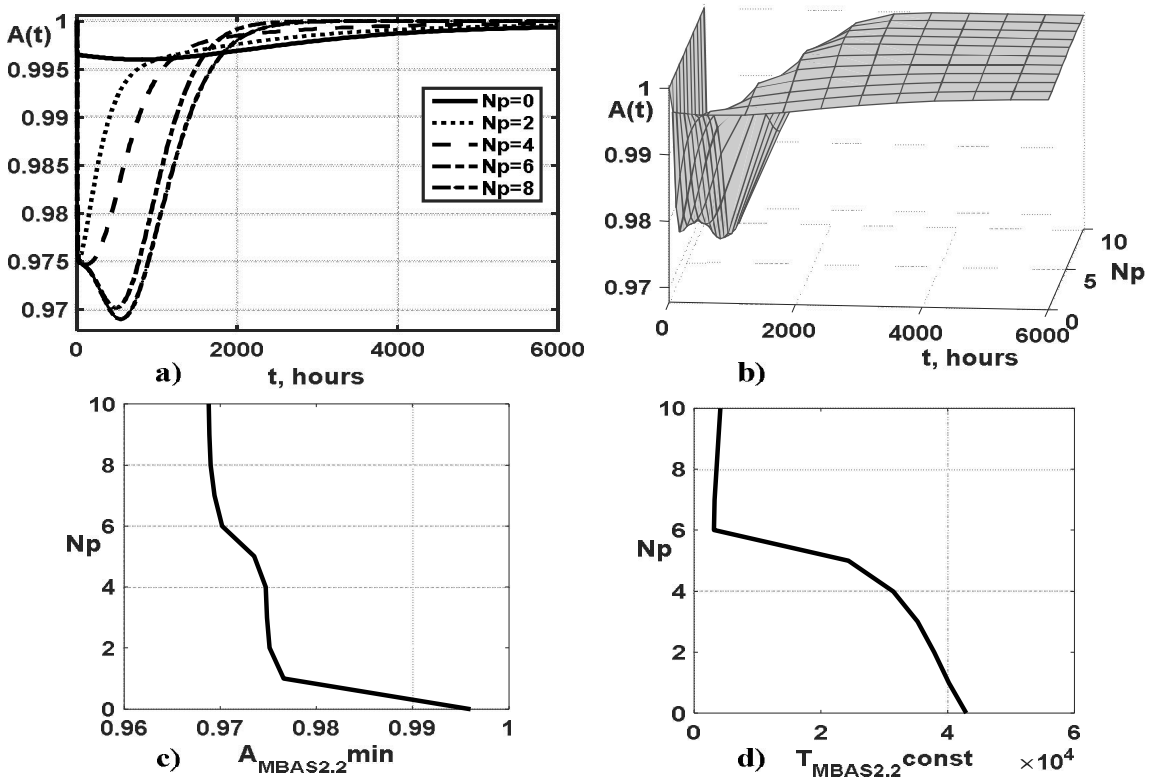

Fig. 5 – The graphs of changing the resulting parameters of the MBAS2.2 model (a, b - availability function, c - minimum of availability function, d - transition period to stable mode with an inaccuracy of 10^{-5}) with a limited number of common maintenances Np.

5. CONCLUSION

In this paper the Markov model architecture is presented with occurred software faults and attacked vulnerabilities considering two maintenance strategies – common and separate.

Comparison of the models with common and separate maintenance strategies allowed to conclude that models MBAS2.1 and MBAS3.1 have higher values of minimums of availability functions (fewer losses of availability in non-stationary mode). The difference between T_{MBAS}const resulting values of MBAS2.1 and MBAS2.2 models is 963.58 hours, the MBAS3.1 and MBAS3.2 models are 1831.6 hours. The transition period for the availability stable mode of the MBAS2.2 model is 4568.2 hours shorter than transition period for the model with common maintenance MBAS3.2; while the elimination of defects and vulnerabilities in model with maintenance is faster than in the MBAS1 model (3.7 times).

Obtained results of simulation strategies can be used for choosing the strategies considering customer requirements. Further steps include:

- development of integrated strategies for BAS maintenance oriented at Cloud Computing taking into account reliability and security policies;
- research of the impact of other types of BAS vulnerabilities on availability and safety.

6. REFERENCES

[1] ISO 16484-1:2010 *Building Automation and Control Systems (BACS) – Part 1: Project Specification and Implementation*, European Committee for Electrotechnical Standardization, Brussels, 2010, 24 p.

[2] ISO 16484-2:2004 *Building Automation and Control Systems (BACS) – Part 2: Hardware*. European Committee for Electrotechnical Standardization, Brussels, 2004, 58 p.

[3] A.-S. M. Q. Abdulmunem and V. S. Kharchenko, "Availability and security assessment of smart building automation systems: combining of attack tree analysis and Markov models," in *Proceedings of the Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, Chania, 2016, pp. 302-307.

[4] A.-S. M. Q. Abdulmunem, W. A.-K. Ahmed, V. Kharchenko, "Ata-based security assessment of smart building automation systems," *Radioelectronic and Computer Systems*, Vol. 3, No. 77, pp. 30-40, 2016.

[5] ISO/IEC 15408-1:2009 *Information Technology – Security Techniques – Evaluation Criteria for IT Security - Part 1: Introduction and General Model*, European Committee for

Electrotechnical Standardization, Brussels, 2009, 74 p.

[6] ISO/IEC 15408-2:2008 *Information Technology – Security Techniques – Evaluation Criteria for IT Security – Part 2: Security Functional Components*, European Committee for Electrotechnical Standardization, Brussels, 2008, 218 p.

[7] U. Farooq, Z. Marrakchi, H. Mehrez, *Tree-based Heterogeneous FPGA Architectures: Application Specific Exploration and Optimization*, Springer, New York, 2012, 188 p.

[8] L. Boyanov, Z. Minchev, "Cyber security challenges in smart homes," in Proceedings of *NATO-ARW "Best Practices and Innovative Approaches to Develop Cybersecurity and Resiliency Policy Framework"*, Ohrid, Macedonia, June 10-12, 2013, pp. 99-114.

[9] Du Suguo, Zhu Haojin, *Security Assessment via Attack Tree Model*, in: Security Assessment in Vehicular Networks, Springer, New York, 2013, pp. 9-16.

[10] M. Grottke, H. Sun, R. M. Fricks, and K. S. Trivedi, "Ten fallacies of availability and reliability analysis," in *Proceedings of the 5th International Conference on Service Availability (ISAS'08)*, T. Nanya, F. Maruyama, A. Pataricza, and M. Malek (Eds.), Springer-Verlag, Berlin, Heidelberg, 2008, pp. 187-206.

[11] T. Huffmire et al., "Managing security in FPGA-based embedded systems," *IEEE Design & Test of Computers*, Vol. 25, No. 6, pp. 590-598, Nov.-Dec. 2008.

[12] D. M. Nicol, W. H. Sanders, K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 1, pp. 48-65, 2004.

[13] M. K. Binu, K. P. Zachariah, "New techniques to enhance FPGA based system security," *International Journal of Advanced Research in Computer Engineering & Technology*, Vol. 1, Issue 5, pp. 91-94, 2012.

[14] X. Ban, T. Xin, "Scenario-based information security risk evaluation method," *International Journal of Security and its Applications*, Vol. 8, No. 5, pp.21-30, 2014.

[15] W. Granzer, W. Kastner, G. Neugschwandtner, F. Praus, "Security in networked building automation systems," in *Proceedings of the IEEE International Workshop on Factory Communication Systems*, 2006, pp. 283-292.

[16] G. Osma, L. Amado, R. Villamizar, G. Ordoñez, "Building automation systems as tool to improve the resilience from energy behavior approach," *Procedia Engineering*, Vol. 118, pp. 861-868, 2015.

[17] S. V. Bhusari, *Smart building integration*, 2014, [Online]. Available: http://www.csemag.com/single-article/smart-building-integration/

[18] E. L. Ler, *Intelligent Building Automation System*, [USQ Project], 2006, [Online]. Available: https://eprints.usq.edu.au/2507/

[19] I. Kuon, R. Tessier. and J. Rose, "FPGA architecture: survey and challenges," *Foundations and Trends in Electronic Design Automation*, Vol. 2(2), pp. 135-253, 2008.

[20] V. S. Kharchenko, A.-S. M. Q. Abdulmunem, Y. L. Ponochovnyi, "Markov availability model of smart building automation system with separate and common reliability-security related maintenance," *Systems of Control, Navigation and Communication*, Vol. 4(36), pp. 88-94. 2015.

[21] ISO/IEC 61508-4:2010, *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. Part 4: Definitions and Abbreviations*, European Committee for Electrotechnical Standardization, Brussels, 2010, 42 p.

***Prof. Vyacheslav Kharchenko**, graduated Kharkov High Command-Engineering Military School of Rocket Forces. Now he works as head of Computer Systems and Networks department at National Aerospace University "KhAI". Research interests: multiversion systems theory; methods and means of assessment and ensuring for reliability, survivability, and functional safety of information control and processing systems; technologies for developing and assessment of dependable systems, aerospace complexes, business-critical systems.*

***Ph.D., Yuriy Ponochovnyi**, graduated Poltava Military Institute of Communications. Now he works as Associate Professor of Computer Engineering Department at Poltava National Technical University. Research interests: methodological bases and IT for reliability, safety and security management of distributed infrastructures.*

***Al-Sudani Mustafa Qahtan Abdulmunem**, graduated National Aerospace University KhAI. Now he works as graduate student of Computer Systems and Networks department at National Aerospace University "KhAI". Research interests: models and methods of information technology for cyber security and availability ensuring of smart building information and control systems.*

***Ph.D., Artem Boyarchuk**, graduated National Aerospace University KhAI. Now he works as Senior Lecturer of Computer Systems and Networks department at National Aerospace University "KhAI". Research interests: Computer Communications (Networks), Software Engineering, Computer Security and Reliability.*