Model of Information and Control Systems in Smart Buildings with Separate Maintenance by Reliability and Security

Vyacheslav Kharchenko[1,3] [0000-0001-5352-077X], Yuriy Ponochovnyi[1,2] [0000-0002-6856-2013], Al-Sudani Mustafa Qahtan Abdulmunem[1] [0000-0002-6297-1983], Aleksandr Ivasiuk[3] [0000-0002-4737-9244], Oleg Ivanchenko[4] [0000-0002-5921-5757]

[1] National Aerospace University KhAI, Kharkiv, Ukraine

V.Kharchenko@khai.edu, mostafahkahtan1@gmail.com

[2] Poltava National Technical University named after Yurij Kondratyuk, Poltava, Ukraine

pnch1@rambler.ru

[3] Research and Production Company Radiy, Kirovograd, Ukraine

ivasiuk.radiks@gmail.com

[4] University of Customs and Finance, Dnipro, Ukraine

vmsu12@gmail.com

Abstract. This article presents the information and control system of smart building is considered as a set of subsystems including a building automation system (BAS). BAS security and availability during its life cycle are assessed using the Markov models. Markov model is used to develop number of strategies which help to recover system and elimination all the possibility threat, during life of systems. Strategies of developing Markov models for describing the recovery of system components after an attack or a software failure are discussed. The use of Markov models is usually justified by the customer's requirements for a specific criterion for assessing the quality of the system.

Keywords: Markov model, building automation system, maintenances strategies

1 Introduction

The development of cloud computing and virtualization technology are responsible for the appearance of new variants of the architecture of IT systems, which include a system of "smart home". IT systems must be considered when assessing and ensuring the quality of modern computer systems and services. 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].

Modification of software tools of different architecture levels of the smart building BAS due to the elimination of design defects and patching of vulnerabilities leads to a change in the parameters of the failure and recovery flows of the system. As it was shown in the works [3,4], it is preferable to use the apparatus of Markov and semi-Markov processes to study systems with variable parameters [5-7]. In [8], a systematic approach to the construction of multi fragment models is developed, and in [9,10], models that take into account reliability and security factors for web systems have been developed. However, in known studies, the influence of different maintenance strategies concerning these factors has not been investigated.

Thus, it is necessary to choose a more acceptable approach for constructing Markov models of BAS availability for separate maintenance, taking into account the gradual elimination of software defects and vulnerabilities.

2 Approach and Modeling Technique

2.1. Building Automation System Architecture and Components

BAS components are different depending on the area of system application, but in general they can be divided as:

1. Upper level (Management Level): dispatching and administration as well as work with databases and statistical functions. At this level cooperation between personnel (operators, dispatchers etc.) and system is performed, which is implemented by means of computer devices and SCADA-systems. In our case study, we analyze the database of this level taking into account reliability and security.

2. Middle level (Communication Level): it is responsible for connection between levels and sending/receiving the information. According to our analysis we choose Wireless Network as one of these level components [11].

3. Low level (Automation Level): level of terminals with input/output functions. This level includes sensors, actuating mechanisms, cabling between devices and low-middle levels. One of the important components used for this level is FPGA [12].

These levels are divided depending on our vision of analysis for the system. There are different designs of BAS but we choose this design as the easiest in use and analysis.

Taking into account the positions of reliability and cyber security allows expanding the list of causes of failures and weaknesses of the system within the framework of a unified dependability concept. In the direction of reliability, hardware and software defects, as well as interaction defects due to operating personnel errors and attacks on the system are analyzed. On the cyber security aspect, software vulnerabilities, Trojans and backdoors are analyzed (Fig. 1).

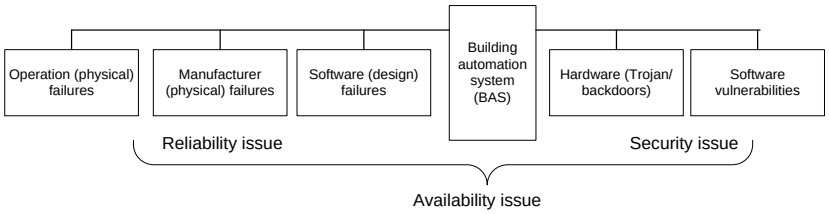


Fig. 1. Analysis availability of building automation system

2.2. Component Faults and Vulnerabilities (Specification)

Our analysis deals with static and not complex system; however, in case we have a big system with different number of components it will be more complicated to use this method. In this step we start to develop Markov model. In the Markov model [3] we have possibility to add more components and eliminate them without any effect on the analysis process. During the first step in analysis process we need to give a big picture for system with all possible sates, which the system can be in throughout its lifecycle in Fig.2.

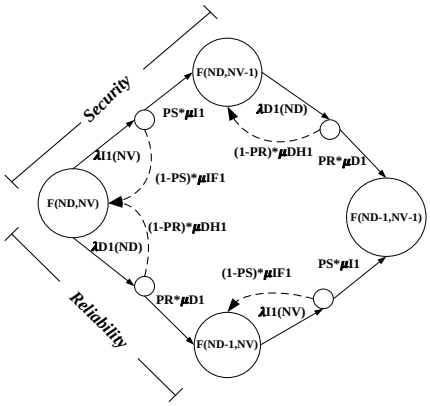


Fig. 2. Markov graph of BAS availability

The Markov model, which analyzes all the possible states of the system and shows the transmission between state and recovery.

2.3. Model Specification (Including Description of Maintenance Strategies)

In this work we develop five models using Markov model as show in Table 1. The BAS analysis is divided into security issues and reliability issues. The states 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 by the minimum time of the

system life cycle, and recover to the maximum value of availability (A_{MBAS} constant) during period of time (T_{MBAS} constant).

Table 1. Basic models BAS

General characteristics of the model	Model specification	Conventional notions
A) Base model without maintenance	-the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances 0	MBAS1
B) Model with common maintenance	- the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances: unlimited during the system whole life cycle - type of maintenance: common	MBAS2.1
	- the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances: 0..Np - type of maintenances: common	MBAS2.2
C) Model with separate maintenance	- the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances: unlimited during the system whole life cycle - type of service: separate	MBAS3.1
	- the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances by defects 0..Ndp, - the number of maintenances by vulnerabilities 0..Ndv - type of service: separate	MBAS3.2

In some cases, the elimination process inside the system will not be able to eliminate the vulnerability or design fault; in this case we add the maintenance strategies, which give the support for system to increase the elimination process. In our case we use two types of maintenances strategies:

1. The common maintenance, which deals with design fault and vulnerability in same time, and it means that 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 will be describing 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 Separate Maintenance

This model describes system functioning in the context of separate maintenance activities, the number of such activities throughout the life cycle is limited.

Simulation shows the principle: at the planning stage of the maintenance procedures, developers can only assume the number of undetected defects and

vulnerabilities. But unlike the common maintenance model, the MBAS3.2 model knows for sure that only vulnerabilities will be fixed during the maintenance of vulnerabilities, and only defects will be eliminated during defect maintenance. Therefore, in the MBAS3.2 model, the Ndp and Nvp input parameters determine the planned number of maintenances for defects and vulnerabilities, respectively.

The marked graph of the model is shown in Fig. 3. When constructing the graph of the model to increase the visibility, it was assumed that the defect or vulnerability was completely eliminated without restarting the system (i.e., PR = PS = 1). But this assumption concerns only the graphic representation in Fig. 3; subsequent simulation results take into account the restart of the system.

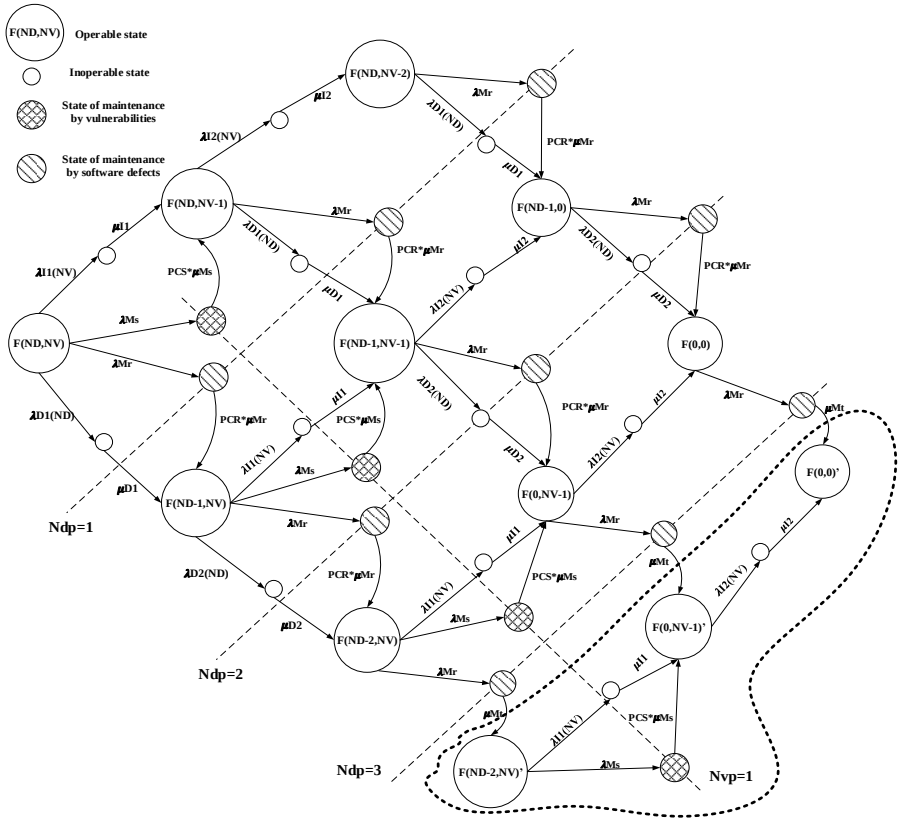


Fig. 3. Marked graph of the MBAS3.2 model taking into account the limited number of separate maintenances by defects (Ndp = 3) and vulnerabilities (Nvp = 1)

The graph in Fig. 3 is the BAS model with two defects and two vulnerabilities (Nd = 2, Nv = 2), and it additionally describes three maintenances by defects (Ndp=3) and one maintenance by vulnerability (Nvp = 1). The planned number of maintenances (for example, over defects) determines not the number of vertical diagonals of the rhomboid Fig.3 of the orgraph, but corresponds to inclined lines in the direction of the shift when eliminating defects (right-down). In detecting and eliminating defects, the logic of the functioning of the MBAS3.2 model is the

following: the first maintenance (Ndp = 1) is performed after the system is put into operation and has three probable states (with transitions from the states F(Nd, Nv), F(Nd, Nv-1)), F(Nd, Nv-2)). After maintenance, the detected defect is eliminated, therefore, the second maintenance (Ndp=2) also has three probable states (with transitions from the states F(Nd-1, Nv), F(Nd-1, Nv-1), F(Nd-1, 0)). Since only two defects were initially present in the system, the third maintenance by defects is redundant and an additional fragment is required for its modeling in the graph (it is shown by a dashed Fig.3 line). The third maintenance also has three probable states.

Since only one maintenance is planned for the vulnerabilities, it will have four probable states with transitions from the states F(Nd, Nv), F(Nd-1, Nv), F(Nd-2, Nv), F(Nd-2, Nv)'. The second vulnerability will be eliminated only after its manifestation.

When building the model, it is necessary to take into account four variants of the forecasting the initial number of defects and vulnerabilities:

a) (Ndp≤Nd)&(Nvp≤Nv); b) (Ndp≤Nd)&(Nvp>Nv);

c) (Ndp>Nd)&(Nvp≤Nv); d) (Ndp>Nd)&(Nvp>Nv).

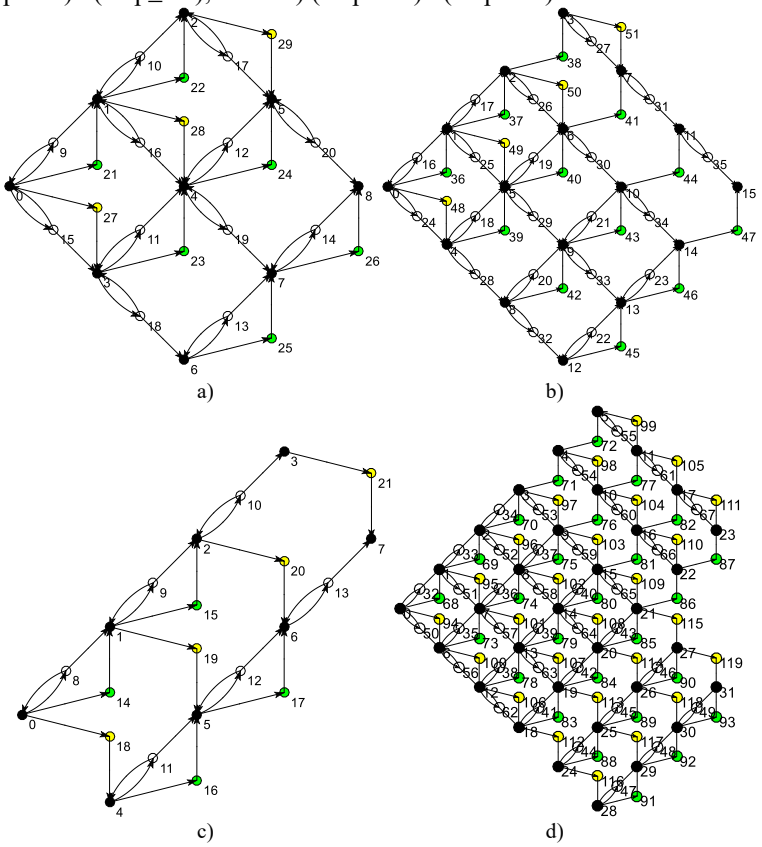


Fig. 4. Marked digraph of MBAS3.2 model taking into account the limited number of separate maintenances for configurations: a) Nd=2, Nv=2, Ndp=1, Nvp=2; б) Nd=3, Nv=2, Ndp=1, Nvp=3; b) Nd=0, Nv=3, Ndp=1, Nvp=2; г) Nd=3, Nv=3, Ndp=5, Nvp=5.

The marked orgraphs of models constructed with these forecast options are shown in Fig.4. Fig.4(a) shows the orgraph of the system with two defects and vulnerabilities, in which the number of maintenances by defects/vulnerabilities does not exceed 2 (two by vulnerabilities and one by defects). To improve the visibility of the state of maintenance over defects are shown in yellow circles, over vulnerabilities – in green. Fig. 4(b) shows the orgraph of the model, in which the predicted number of maintenance by vulnerabilities exceeds their number in the system. This causes the occurrence of additional operable (S3, S7, S11, S15) and inoperable (S27, S31, S35, S51) states. Fig. 4(c) shows the orgraph of the model, in which defects are absent, but one maintenance is planned to be according to defects. This causes the occurrence of additional operable (S4, S5, S6, S7) and inoperable (S11, S12, S13, S16, S17) states. The orgraph of the MBAS3.2 model, in which the number of planned maintenances by both defects and vulnerabilities (Ndp = 5, Nvp = 5) exceeds their real number in the system (Nd = Nv = 3) and is shown in Fig.4(d). After the elimination of all defects and vulnerabilities, the maintenance procedures are carried out for two more periods, and then terminated. In this regard, the availability function covers additional states and is calculated as:

$$A(t) = \sum_{i=0}^{Nk} P_i(t); \tag{1}$$

$$Nk = (Nd+1)(Nv+1)+(Nd+1)(\max(Nvp,Nv)-Nv)+(Nv+1)(\max(Ndp,Nd)-Nd)$$

4 Simulation and Comparative Analysis

The calculation of the availability indicators is performed for the input data from Table 2. To construct the matrix of the Kolmogorov-Chapman system of differential equations, we use the matrixA function [4]. The Kolmogorov solution was performed in the Matlab system using the ode15s method for the time interval of [0 ... 50000] hours. The availability function is determined by (1). The results of the solution are presented in the graphical form in Fig. 5.

The analysis of the graphs in Fig. 5 showed that the limitation of the number of maintenances in MBAS2.2 and MBAS3.2 models makes it possible to achieve an ideal availability (AMBAS2.2const = AMBAS3.2const = 1) in the stable (stationary) mode. The minimum of availability function for models with limited and unlimited maintenance varies:

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

The transition period for the stable mode for MBAS2.2 model is 2.5241 times higher than 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).

Since 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 TMBAS2.2const is considered (in addition, their influence on AMBAS2.2min is analyzed). The dimensionality of the model is increased to Nd = 3, Nv = 3. The Np parameter varies from 0 to 10.

Table 2. 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	
Nd	The number of defects in the system BAS	2	
Nv	The number of vulnerabilities in the system BAS	2	
Np	The number of common maintenance	6	
Ndp	The number of the maintenance separate in defects	2	
Nvp	The number of the maintenance separate in vulnerabilities	2	

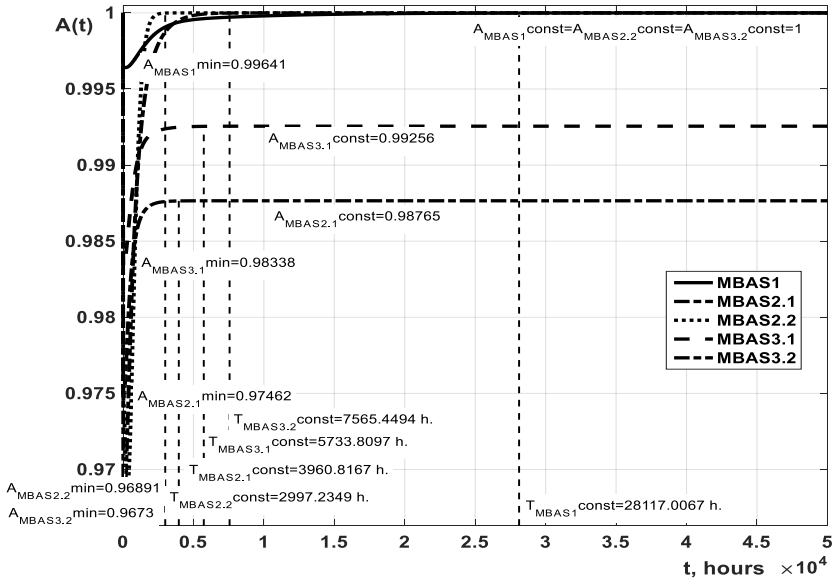


Fig. 5. Preparedness simulation results BAS architecture (the resulting figures are determined with an accuracy of 10^{-5})

The analysis of the graphs in Fig. 5 showed that limiting the number of separate maintenances in the MBAS3.2 model (as in the MBAS2.2 model) allows achieving an ideal availability (AMBAS3.2const=1) in the steady. Also as in the previous MBAS2.2 model, the minimum availability value for models with limited and unlimited maintenance differs insignificantly (by 9.73e-5). However, common maintenance remains an advantageous one according to the AMBASimin (by 0.022) indicator.

If we compare models with limited and unlimited maintenance, then it is clear that the latter has a shorter period of transition of the availability function to the steady state. The difference between the resulting TMBAS iconst indicators of models MBAS3.1 and MBAS3.2 is 882.6 hours. The transition period for the availability function to the steady state in the MBAS3.2 model is 1346.4 hours less than in the limited common maintenance MBAS2.2. In addition, eliminating defects and vulnerabilities in the model with maintenance is faster than in the MBAS1 model (4.2 times).

Since interest is caused by a decrease in the detection and elimination of all defects and vulnerabilities, then further we consider the influence of individual input parameters on the resulting indicator TMBAS 3.2const (in addition, their impact on AMBAS 2.2min is analyzed). The dimensionality of the model is increased to Nd = 3, Nv = 3.

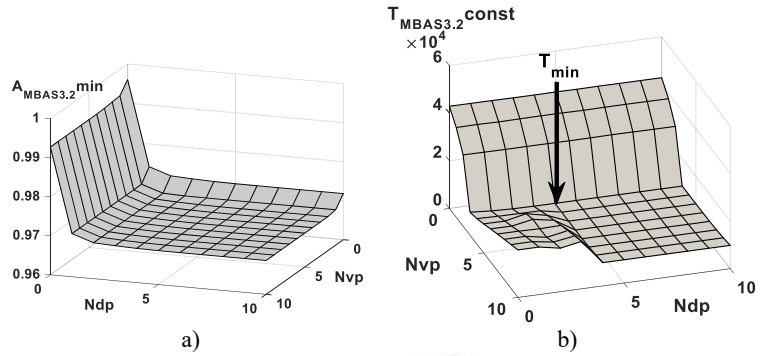
Table 3. The boundaries of the MBAS3.2 model input values

Name	Mathlab-name	Value row	Measur.unit
Predicted number of separate maintenances	Ndp, Nvp	[0..10]	
The intensity of defect detection and elimination μMr	muMr	[0.1..1]	1/hour

The results of modeling in the form of graphical dependencies are shown in Fig.6 – Fig. 8.

Dependence of the resulting indicator AMBAS3.2min on the number of separate maintenances is shown in Fig. 6(a). Analysis of the three-dimensional graph allows to distinguish the following points. The BAS system without maintenance is optimal according to the criterion AMBAS3.2min–>max (Ndp=Nvp=0, AMBAS3.2min=0,996). The system without maintenance by defects (Ndp = 0, Nvp>0) exceeds the system without maintenance by vulnerabilities (Nvp = 0, Ndp>0) by AMBAS3.2min by 0.021. In BAS systems with the number of limited separate maintenances greater than the real number of defects and vulnerabilities (Ndp> 3, Nvp> 3), the change in AMBAS3.2min does not exceed 6.3e-8.

Fig. 6(b) shows the dependence of the transition period of the MBAS3.2 availability function in the steady state on the number of separate maintenances. The location of the minimum on the three-dimensional graph is shown by a special metrics and corresponds to the value min(TMBAS3.2const)=8496,153 hours under the configuration of the number of maintenances Nvp = 3, Ndp = 4. In BAS systems with the number of limited separate maintenances greater than the actual number of defects and vulnerabilities (Ndp> 3, Nvp> 3), the change in the TMBAS 3.2const does not exceed 1256.546489 hours, but there is a growing trend of TMBAS 3.2const with an increase in Nvp, which is shown in Fig. 7.



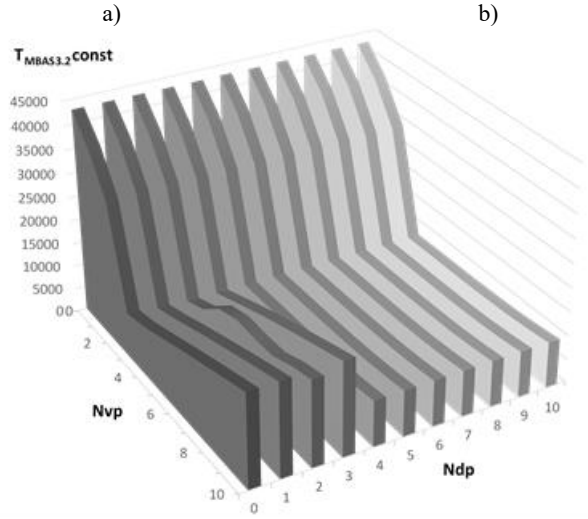


Fig. 6. Graphs of the change in the resulting indicators of the MBAS3.2 model (a – the minimum of the availability function, b – the period of transition to the steady state with the error of 10^{-5}) with a limited number of separate maintenance

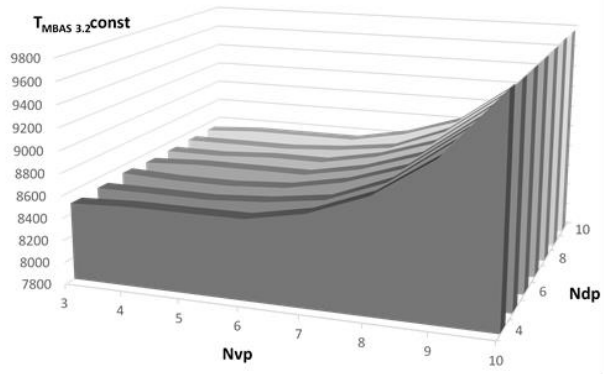


Fig. 7. Details of the change of TMBAS 3.2const in the MBAS3.2 model on the intervals Ndp>3, Nvp> 3

When analyzing the three-dimensional graph in Fig. 6, and over Ndp = const, an insignificant chaotic change in the parameter TMBAS 3.2const is observed at the intervals Nvp<3 and Nvp> 3 under Ndvp> 3 and for the entire interval Nvp = [0..10] under Ndvp< 3. This is shown in detail in Fig. 8.

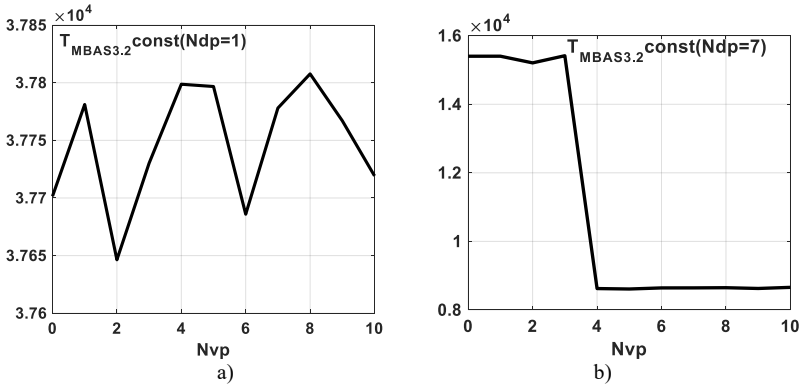


Fig. 8. Detailization of the change in TMBAS 3.2const of the model MBAS3.2 on slices Ndp = 1 (a), Nvp = 7 (b)

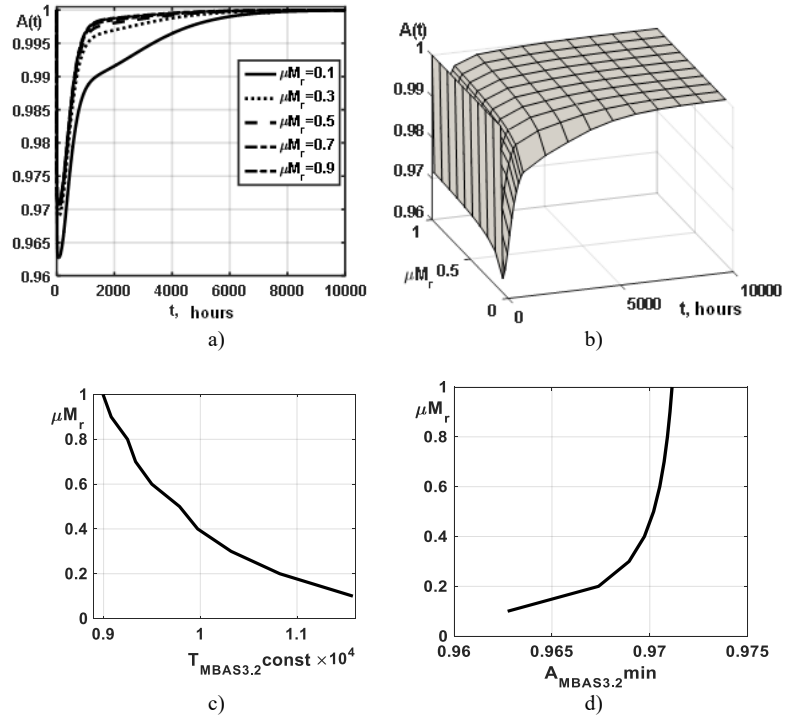


Fig. 9. Graphs of the change in the resulting indicators of the MBAS3.2 model (a, b – availability functions, c – minimum availability function, d – transition period to the steady with the error of 10-5) from the intensity of detection and elimination of the defect μMr

Explanation of this dependence follows from the difference in the input parameters λMs and λMr – with their accepted values (λMs = 5e-3 and λMr = 1e-3), the transition to the maintenance state by vulnerabilities is performed with greater intensity.

Next, the influence of the intensity of the detecting and eliminating the μMr defect on the resulting parameters of TMBAS3.2const and AMBAS3.2min is considered. When constructing models, the values of the input parameters Nv = Nd = 3, Nvp = 3, Ndp = 4 were taken.

The results shown in Fig. 9 also show the expected result: if the maintenance quickly identifies and corrects defects, then the minimum availability function (AMBAS3.2min) increases, and the transition period to the steady state decreases. Thus, with a 10-fold acceleration of detection and elimination of defects during maintenance, the value of AMBAS 3.2min increases by 0.0084, and the period of detection and elimination of all defects and vulnerabilities decreases by 1.2872 times.

5 Conclusions

In the article, the Markov model architecture is presented with occurred software faults and attacked vulnerabilities considering separate maintenance strategies.

Analysis of the obtained results of modeling the availability of the BAS architecture with procedures for common and separate limited maintenance has shown that:

a) limiting the number of maintenance activities allows to increase the value of the availability function in the steady state to one, while the value of the minimum of the availability function remains at the level of systems with an unlimited number of maintenances;

b) the duration of transition of the availability function to the steady state for the MBAS2.2 model is 9.48 times greater than for the model with unlimited common maintenance MBAS2.1; but the elimination of defects and vulnerabilities in the serviced model is faster than in the MBAS1 model (1.27 times);

c) the duration of transition of the availability function to the steady state in the MBAS3.2 model is 1346.4 hours less than for the model with limited common maintenance MBAS2.2; while eliminating defects and vulnerabilities in the maintenance model is faster than in the MBAS1 model (4.2 times).

Result of simulation the strategies can be used for choosing the strategy considering customer requirements. Future 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.

References

1. Europe Smart Homes, BSRIA Worldwide Market Intelligence, https://www.bsria.co.uk/market-intelligence/market-reports/publication/europe-smart-homes-market-2013/.
2. Moreno, M., Úbeda, B., Skarmeta, A., Zamora, M.: How can We Tackle Energy Efficiency in IoT BasedSmart Buildings?. Sensors. 14, 9582-9614 (2014).
3. Abdulmunem A. S. M. Q. and Kharchenko V. S.: Availability and Security Assessment of Smart Building Automation Systems: Combining of Attack Tree Analysis and Markov Models. In 2016 Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), Chania, pp. 302-307 (2016)
4. Kharchenko, V., Ponochovnyi, Y., Abdulmunem, A., Andrashov, A.: Availability Models and Maintenance Strategies for Smart Building Automation Systems Considering Attacks on Component Vulnerabilities. Advances in Dependability Engineering of Complex Systems. 186-195 (2017).
5. Trivedi, K., Kim, D., Roy, A., Medhi, D.: Dependability and security models. 2009 7th International Workshop on Design of Reliable Communication Networks. (2009).
6. Yu, Q., Johnson, R.: Smart grid communications equipment: EMI, safety, and environmental compliance testing considerations. Bell Labs Technical Journal. 16, 109-131 (2011).
7. Osma, G., Amado, L., Villamizar, R., Ordoñez, G.: Building Automation Systems as Tool to Improve the Resilience from Energy Behavior Approach. Procedia Engineering. 118, 861-868 (2015).
8. Kharchenko, V., Odarushchenko, O., Odarushchenko, V., Popov, P.: Selecting Mathematical Software for Dependability Assessment of Computer Systems Described by Stiff Markov Chains. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI-2013, CCIS, vol. 1000, pp.146--162. Springer, Heidelberg (2013)
9. Kharchenko, V., Abdul-Hadi, A., Boyarchuk, A., Ponochovny, Y.: Web Systems Availability Assessment Considering Attacks on Service Configuration Vulnerabilities. Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland. 275-284 (2014).
10. Hafezian Razavi, S., Das, O.: Security Evaluation of Layered Intrusion Tolerant Systems. Analytical and Stochastic Modeling Techniques and Applications. 145-158 (2010).
11. Loukas, G., Gan, D., Tuan Vuong: A taxonomy of cyber attack and defence mechanisms for emergency management networks. 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). (2013).
12. Farooq, U., Marrakchi, Z., Mehrez, H.: Tree-Based Application Specific Inflexible FPGA. Tree-based Heterogeneous FPGA Architectures. 123-151 (2012).