Multiversion environment creation for control algorithm execution by autonomous unmanned objects

Kovalev I V\(^1\), Zelenkov P V\(^2\), Losev V V\(^3\), Kovalev D I\(^4\), Ivleva N V\(^5\), Saramud M V\(^6\)

\(^1\)Dr. Sc., Professor of System analysis department, Rector, Reshetnev Siberian State Aerospace University, 31 “Krasnoyarsky Rabochy” Av., 660037 Krasnoyarsk, Russia.  
\(^2\)Cand. Sc., Vice-Rector for Informatization and Commercialization of Scientific Research Projects, Reshetnev Siberian State Aerospace University, 31 “Krasnoyarsky Rabochy” Av., 660037 Krasnoyarsk, Russia.  
\(^3\)Cand. Sc., Deputy Vice-Rector for Informatization and Commercialization of Scientific Research Projects, Reshetnev Siberian State Aerospace University, 31 “Krasnoyarsky Rabochy” Av., 660037 Krasnoyarsk, Russia.  
\(^4\)Postgraduate student, Reshetnev Siberian State Aerospace University, 31 “Krasnoyarsky Rabochy” Av., 660037 Krasnoyarsk, Russia.  
\(^5\)Senior English Instructor, postgraduate student, Reshetnev Siberian State Aerospace University, 31 “Krasnoyarsky Rabochy” Av., 660037 Krasnoyarsk, Russia.  
\(^6\)Engineer, Reshetnev Siberian State Aerospace University, 31 “Krasnoyarsky Rabochy” Av., 660037 Krasnoyarsk, Russia.

E-mail: basilos@mail.com, natalie_ivleva@mail.ru

Abstract. The article describes one of the approaches to increase resilience of software control systems for autonomous unmanned objects based on the multiversions methodology.

The task of piloting autonomous unmanned objects (space vehicles, including small spacecraft, the CubeSat, nano-, picosatellites, drones, quadrocopters, unmanned ground vehicles) is associated with a number of functions: providing telemetry; development of control actions; maintaining autonomy, including energetical, resource and functional one, which, in turn, imposes a number of higher requirements to the object to increase fault tolerance [1], as well as the ability to perform the functions assigned to it for the maximum operational period available. The issue of increasing resiliency of objects of this class goes beyond the traditional “fault-tolerant computing” concept based on the introduction of the hardware redundancy in the system, for example, because of weight restrictions in the case of orbital objects. Thus, the focus of research is shifting towards software and software failures, namely, in the application of the multiversional programming aiming to develop design procedures failover software system of autonomous unmanned objects. The definition of theoretical and practical tools of choosing the multiversion model that meets the requirements, and the method of multiversion voting enabling at the stage of design and management systems modeling, as part of on-board equipment of autonomous unmanned facilities, taking into account its purpose and functioning, will improve the resilience of software systems of autonomous unmanned objects.

The main objective of the study is the use of the design methodology failover software to improve the quality of the algorithms in autonomous unmanned objects [2].

Given the technological complexity of the production cycle of final products as well as resource cost, the including temporary one, the task of ensuring the reliability and resiliency of autonomous unmanned aerial objects is of utmost importance. Since tasks, assigned to the spacecraft, become more complex and the number is growing in the algebraic progression, while the share of self-reliance features preserved and
functional gains achieved through payload (telecommunications, scientific and other equipment), the issue of failover control software systems of autonomous unmanned vehicles, as a result of the security and viability of objects and infrastructure formed by them [3].

Methodological basis of multiversion environment for autonomous unmanned objects algorithm performance uses multiversion technology to improve reliability of the software system of autonomous unmanned objects management system that allows you to analyze multiversion model of selective algorithm and their attribute differences to achieve the specified reliability characteristics of the software module designed [4].

This formed the basis of the methodology designed to evaluate the effectiveness of multiversion models. This method implements the step by step mechanism for determining the value of the relationship between the correctness of Sn and the number of the multiverse n.

This method has an implementation model for recovering the blocks (DHS), and for multiversion program (IMP).

The main steps implementing this technique are:

a. classification of models, proposed for use in multiversion runtime;

b. situational probability estimation errors or return the correct result of software modules and a decision or an error verification module to DHS;

c. Review and assessment of exceptions to the DHS;

d. probabilistic assessment return errors or correct result of executable modules and a decision or an error of the algorithm for the MVP voting.

The following is an example of a technique for EBM and IMP.

Model recovering unit receives a “correct” result, when at least one of the multiverse returns “valid” and the result of the verification module receives the correct decision.

Suppose, we have n versions and their reliability is r1, r2, ..., rn respectively. Reliability verification module is b. Let’s define the relation between the correctness of the system output (Sn) and the number of the multiverse (n). The following expression for Sn:

\[ S_n = S_{n-1} + [r_1(1-b) + (1-r_1)b] \cdot ... \cdot [r_{n-1}(1-b) + (1-r_{n-1})b] \cdot r_nb \] (1)

If all modules have a certain average reliability:

\[ S_n = S_{n-1} + [(1-b) + (1-b)] \cdot n^{-1} \cdot b \] (2)

Let us consider some special cases.

1) The reliability of the inspection module is equal to (b = l), then

\[ S_n = 1 - (1^-r)^n \]

Note that Sn→1, when n→∞. In this case, we have formed a system to ensure guaranteed reliability which depends proportionally on their amount.

2) If average reliability software module is equal to one (= 1), then

\[ S_n = 1 - (1-b)^n. \]

As in the previous case, Sn→1 for n→∞. In this case the reliability of the verification module does not matter, as long as the amount of the multiversions was large enough.
By using the methodology multiversion programming system reliability is mostly determined by the algorithm used by the multiverse matching results. The voting algorithm used in the theoretical study of the limit reliability and able to evaluate all the possible conditions that affect the output of the system is the algorithm of the vote by the absolute majority.

The vote by the absolute majority received the “correct” result, if more than half of the multiverse return “correct” the result of the vote and the algorithm selects one of them. Assuming that all program modules have a certain mean reliability, the reliability decision module - b, and the correctness of the output system - Sn, where n - the number in the multiverse system.

Assuming returned the “correct” result of m modules, and (n-m) - returned an error, then the probability of such a situation can be written as:

\[ P_n(m) = C^m_n \bar{r}^m (1-\bar{r})^{n-m}. \]

Then Sn can be represented as the sum of the probabilities that m = \[ \left\lfloor \frac{n}{2} + 1 \right\rfloor \] modules returned “right” result and the voting algorithm made the right decision:

\[ S_n = b \sum_{j=m}^{n} C^j_n \frac{a_j}{b_j} (1-\bar{r})^{n-j}. \]

It is also possible to estimate the ultimate reliability of the system through the improvement of all kinds used by the voting algorithm.

Our procedure allows to evaluate the effectiveness of the main multiversion models on different n - multiverses. The term “correctness” is an important property in the context of the development of fault-tolerant software. Higher the value is, more reliable the system is. This property has a higher priority than the availability and reliability, as if we know the value of the “correctness”, we can say with certainty how many results (multiverse outputs) contains no errors and only use them, thus increasing the accuracy of the system. And higher the accuracy of the system is, higher the availability and reliability are.

For modified versions of multiversion voting algorithms aimed at improving the stability of algorithms of multiversion errors of software modules (compared to the unweighted analogues) the key element of multiversion system is a decision block. This block divides outputs of multiple software versions on the “correct” and “incorrect”. There are several methods of separating of outputs, while the most common of them are based on the classification of outputs, so the most promising of these techniques are voted by an absolute majority, a vote agreed by a majority, and fuzzy vote with majority agreement. As part of this research, a number of improvements are proposed to these techniques, namely, the weighted vote majority consistent and fuzzy weighted majority voting, based on the statistics of the program modules. The basic idea is that each module has a certain weight (a number from 0 to 1) during decision making. As a software weight the probability of return “correct” answer is used.

Functional expansion of the methodological framework of multiversion programming will deliver the reliability characteristics of software modules, and hence the predictability of their performance in multiversion environment [5].

Model-algorithmic component of multiversion environment autonomous unmanned facility management algorithms execution is formed from:

1. Algorithm of selecting multiversion vote methods to measure the effectiveness of algorithms. The study will be presented with an option to choose the method by voting multiversion by simulating the characteristics of the real system in a special simulation environment. The purpose of this
environment is to quantify the effectiveness of the method of voting multiversions used in the decision-making unit multiversion system, in particular a system with specified characteristics (number of modules, the average unit reliability, the probability of multiversional errors).

2. Software tools implementation (multiversion runtime), allowing to unify application multiversion approach to the various software complexes, which, unlike currently available allow executables are not only using multiversion programming methodology, but also with the other common multiversion models: recovering blocks in recovering units, \( t / (n-1) \) - versioned programming and multiversion programming with self-examination, etc. The principle of duality, laid in multiversionnyh execution environment autonomous unmanned facility management algorithms, is designed to provide not only the fault tolerance software management system kernel, but also to improve the accuracy of processed data results - algorithms provide telemetry [6].

Thus, the formed methodological basis is intended to not only provide a theoretical estimate of the limit of reliability of software modules, but also to perform model-algorithmic component in the design and implementation of software multiversion runtime.

References

[1] Kuznetsov PA, Kovalev IV, Losev VV, Kalinin AO, Murygin AV 2016 To the question about the states of workability for automatic control systems with complicated structure IOP Conf.Series: Materials Science and Engineering vol.45

[2] Kovalev IV, Zelenkov PV, Karaseva MV, Kovalev DI Ant algorithm modification for multi-version software building. Lecture notes in computer science. Springer-Verlag. GmbH. vol. 1940, 2015, 222-228 p.

[3] Engel EA, Kovalev IV 2011 Information processing using intelligent algorithms by solving WCCI 2010 tasks Vestnik SibGAU vol. 3, no. 36, pp.45

[4] Kovalev IV, Zelenkov PV, Ognerubov S. The minimization of inter-module the interface for the achievement of reliability of multi-version software IOP Conference Series: Materials Science and Engineering.(XVII International Scientific Conference «Reshetnev readings »), vol.45

[5] IV of Kovalev, Kovalev the DI, the PV Zelenkov, Voroshilova the AA of The Mathematical statement for the solving of the problem of of N-version software system design IOP Conference Series: Materials Science and Engineering Ser."International Scientific and Research Conference on Topical Issues in Aeronautics and Astronautics (Dedicated to the 55th Anniversary from the Foundation of SibSAU)" in 2015. S

[6] Raspopin NA, Karaseva MV, Zelenkov PV, Kayukov EV, Kovalev IV 2012 Models and methods of optimization of information acquisition and processing Vestnik SibGAU vol. 2, no. 42 pp.45

“The reported study was funded by Russian Foundation for Basic Research, Government of Krasnoyarsk Territory, Krasnoyarsk Region Science and Technology Support Fund to the research project № 16-47-242143”