Choice of optimal multiversion software for a small satellite ground-based control and command complex

I N Kartsan, S V Efremova, V V Khrapunova and M I Tolstopiatov
Reshetnev Siberian State Aerospace University, 31 Krasnoyarsky Rabochy ave., Krasnoyarsk, 660037, Russia
E-mail: kartsan2003@mail.ru

Abstract. This paper discusses the criteria and restrictions, which arise during the solution of problems relevant to the synthesis of software for a ground-based control and command complex for small satellites. The authors suggest a number of models for forming multiversion software for this complex.

1. Introduction
The multiversion methodology for designing software for a ground-based control and command complex for small satellites (GCCCS) proposes its modular composition. In the general case, when designing the GCCCS, it is necessary to create (in accordance with the accepted criterion of efficiency) an optimal software system with two types of restrictions [1].

The first type of restrictions characterizes the level of modern knowledge of the theory and methods for solving the set tasks; principles for constructing the basic functional algorithms; methods for the structural construction of complex systems and the technology for their design.

The second type of restrictions relates mainly to the technical parameters of the means for which the complex system is to be implemented, and to the resources that can be allocated for the development and operation of the system. When designing a software system, such technical constraints, first of all, are the parameters of the computer on which the software system is supposed to be implemented. In addition, the resources of design are the material and financial costs that are available both in the process of designing and during the subsequent operation of the system.

The use of the modular principle at the stage of technical design is associated with the process of optimizing the composition and interrelationships of the individual components of the software and information support of the control and command complex, which has optimal characteristics associated with the development, debugging, and operation of the GCCCS [1–2].

The main criteria for the synthesis of modular GCCCS at the stage of technical design are: (1) the minimum complexity of the intermodule interface; (2) the minimum exchange time between the operational and external memory of a computer when solving problems; (3) the minimum amount of unused data when sending between operational and external computer memory; (4) minimum technological complexity of data processing algorithms; (5) maximum informational performance of the control and command complex; (6) and, in solving problems, maximum reliability of data processing [3].

As constraints for solving the problems of the synthesis of an optimal modular GCCCS, other characteristics are used, such as the composition of procedures and the volume of each module, the composition and volume of information arrays, the complexity and composition of the interface
between the individual modules, the degree of duplication of procedures and information elements, the
volume computer memory, the number of communication channels with external storage devices, the
ability to transfer control within and between modules, the sequence of procedures, spent costs and
time for the development and operation of the modular control and command complex, etc.

The initial data for solving the problems of the synthesis of an optimal modular control and
command complex are the terms of reference for the development of the system, the results of the
analysis of the existing complexes and the results of its preliminary study.

Based on this data, the following characteristics are determined [4]: (1) a set of functional data
processing tasks; (2) a variety of data processing procedures, including alternative ones; (3) a set of
information elements of the system related to data processing procedures and identified by types –
input, intermediate, output; (4) options for the possible interaction of data processing procedures with
information elements; (5) technological adjacency matrix, or technological skeleton matrix; (6) a set of
valid sequences of procedures; (7) characteristics of the procedures, information elements and
technical means.

The optimality of the parameters of the synthesized modular GCCCS is determined by the value of
the specified attributes. The admissibility of various options is determined by the restrictions on the
characteristics of the system being developed.

In the synthesis of the optimal modular GCCCS, its various attributes are used: (1) development
and implementation costs; (2) development and implementation time; (3) operating costs; system
losses due to late submission of data processing results; (4) system speed (system response time); (5)
the required amount of memory for the placement and storage of information arrays and software
modules; (6) the number and load of technical equipment used; (7) accuracy of the processed
information; (8) system reliability; etc. For the designed system, one of the above attributes is used as
the optimality criterion of the system being synthesized, while others are restricted [5]. For this, an
iterative algorithm has been developed for dropping versions of multiversion software modules on
attribute restrictions. The scheme of the iterative algorithm is as follows.

First, all versions of each module are ordered by the values of the attribute being analyzed. The
tolerance is determined, that is, the difference between the restriction on the software being created
and the minimum attribute values of the module versions. After that, the versions which do not pass
the approval are “screened out” and, therefore, cannot be used in the formation of multiversion
software. All other attributes are treated in the same way.

The algorithm completes its work if the dropout no longer occurs, that is, the composition of the
multiversion software is selected, or if all versions of one of the modules are dropped. In the latter
case, you must use less resource-intensive versions of the modules or extend restrictions [6].

One of the main criteria for determining the quality of the developed modular GCCCS is the
complexity of inter-module informational links (inter-module interface), which dominates in
developing, debugging, implementing, and modifying software, or some of the second group criteria,
which is dominant in solving data processing problems.

The criteria of the second group are determined by the performance of the computing facilities used
and the time of exchange with external computer memory, which, in turn, depend on the type and
amount of memory used, the structure of software modules and information arrays, their relationship
to the data processing system, the sequence of problem solving and the implementation of individual
modules, etc.

Another main criterion for the synthesis of an optimal modular GCCCS, which dominates the
solution of functional problems, is the minimization of total time with external computer memory.

Let us consider in more detail the methods for determining the aforementioned basic criteria of
optimality and the peculiarities of their use in the technical design of an optimal modular GCCCS.

Program modules are connected with general input data, as well as data which is the result of the
implementation of some programs and the source for others. Usually, information elements are
combined into arrays according to the signs of uniformity, features of production and use, placement
in memory, belonging to objects, etc., playing the role of an information interface between program
modules. At the same time, the large number and complex nature of informational links between program modules increases the number of errors in the design of software and information support, the effect of errors in a separate program on the work of other programs, complicates the process of separate design of individual parts of the system.

Therefore, in the design process, the number of such connections (its interface) must be minimized with restrictions on the total time for solving data processing problems.

The need to minimize the interface is also caused by the fact that its testing in complex data processing systems requires more than half of the total development costs.

If there are difficulties in estimating the temporal characteristics of accessing the program modules and information arrays of the system, it is expedient to state the problem of synthesizing the optimal modular GCCCS link for the minimum number of calls to external memory during its operation. Indicators of the quality of the synthesized optimal modular GCCCS associated with the time of obtaining the output results are the minimization of the number of transfers of unused procedures and data when solving a problem.

Combining procedures into modules and data into arrays, taking into account the allowable amount of RAM, which should ensure the storage of software modules and related data, it is possible to significantly reduce the number of accesses to external memory. This characteristic of the designed modular GCCCS is estimated using the indicator, which is called the technological complexity of the data processing algorithm (this is the ratio of the total volume of procedures and data transfers when solving a problem to the volume of useful input data transfers of the problem being solved). Minimizing the technological complexity of the modular algorithms of the GCCCS allows one to reduce the number of accesses to external computer memory and the amount of intermediate data.

From the point of view of the users of the modular GCCCS, information performance is one of the most important attributes of their quality assessment. It is defined as the ratio of: (1) the amount of useful data transfers when receiving specified output results to the total time they were received (or the total time to access the external memory when solving a problem); (2) the average amount of information in the elements of the output data stream of the control system and information processing to the average time of their input.

A necessary level of effectiveness of the functioning of the GCCCS is the precision of the data received by its users.

As constraints for solving the problems of synthesis of an optimal modular GCCCS, such characteristics as the composition of procedures and the volume of each module, the composition and volume of information arrays, the complexity and composition of the interface between modules, the degree of duplication of procedures and information elements can be used in addition to the above attributes.

Typical are the technological restrictions on the computer’s RAM, the number of channels, the ability to transfer control between inside and between modules, allowable costs and time to develop and implement modular software of the GCCCS, etc

2. Basic models of the formation of the multiversion OCC ICA

Previously, many authors [7–10] showed the application of well-known optimization models to the problems of determining the optimal level of redundancy in multiversion software systems. The basic information used regarding the reliability and cost indicators of the module also made it possible to establish a relationship (or choose) between these two factors [11–12].

Below are four models, each of which is applicable to different structures of the multiversion software of the GCCCS (from a very simple structure to a more complex one) [13]. The variety of models gives the designer the flexibility to choose the appropriate model for the designed system software.

Let us consider the structure of the models and present the problem statement and solution methods for each of them, by entering the following values:

\[ K \] is the number of functions of the multiversion GCCCS of a communication satellite system that
should be realized;
$F_k$ is the frequency of employing the $k$-th function of the software, $k = 1, 2, ..., K$;
$n$ is the number of multiversion GCCCS modules;
$m_i$ is the number of $i$-the module versions, $i = 1, ..., n$;
$R$ is the probability of the multiversion GCCCS performance without malfunctioning;
$R_i$ is the probability of the $i$-th module performance without malfunctioning;
$R_{ij}$ is the performance probability of the $j$-th version of the $i$-th module without malfunctioning;
$Z_{ij}$ is the Boolean variable equal to 1 if the $j$-th version is selected for the $i$-th module; otherwise it is equal to 0;
$C_{ij}$ is the cost of designing and servicing the $j$-th version of the $i$-th module;
$B$ is the cost restriction for designing a multiversion GCCCS.

The basic assumptions of the multiversion software design methodology are valid, concerning, firstly, the use of modular programming methods; secondly, the independence of designing module versions and the feasibility of evaluating their reliability and cost using COTS support modules (these modules are designed and tested independently, so their reliability can be assessed using any of the available models, and their actual cost is the cost of procurement and maintenance); thirdly, there is a limit on the cost of the multiversion software of the GCCCS.

Let us formulate the basic models for various structures of the multiversion software of GCCCS systems, which allow determining the optimal method of multiversion software while maximizing software reliability with constraints on costs.

The SNR model forms the optimal composition of the modules of the single-function multiversion software of the GCCCS without introducing software redundancy. Multiversion software consists of a single program that performs one “main” function. The program is composed of a set of modules for sequential execution. More than one version of each module is available, but due to tight cost constraints (this factor should not be excluded from consideration) and / or the non-critical nature of some or all of the software system, saving multiple module versions is undesirable or impossible. The model developed for this situation allows for the optimal composition of a set of modules for a single program, maximizing reliability, with the existing cost constraints, due to which the full cost of development remains acceptable.

The target function:

$$\max_{x_1} R = \max_{x_1} \left\{ \prod_{i=1}^{n} \sum_{j=1}^{m} Z_{ij} R_{ij} \right\}, \quad (1)$$

in accordance to the restriction:

$$\sum_{j=1}^{m} Z_{ij} = 1, \quad i = 1, \ldots, n; \quad (2)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} Z_{ij} C_{ij} \leq B \quad; \quad (3)$$

where $Z_{ij} = 0, 1; i = 1, \ldots, n; j = 1, \ldots, m_i$.

The objective function of the SNR model reflects the sequential execution of the modules. The combination of constraints (2) ensures that one version is necessarily selected for each module. Constraints (3) ensure that the total costs do not exceed B. This task corresponds to the problem of nonlinear integer programming.

It should be noted that problem can be solved if
\[ \sum_{i=1}^{n} (\min_{i} C_{ij}) \leq B. \]

Let \( Z_i \) corresponds to the module selected for the problem \( i, (i = 1, ..., n) \), let \( Z_i^o, i = 1, ..., n \) is an optimal solution of a whole functional, when (3) is ignored. Evidently, if \( (Z_1^o, ..., Z_n^o) \) is the optimal solution, there must be an optimal solution for the entire target function

\[ \prod_{i=1}^{n} (\max R_{ij}) \]

Without loss of generality, we can assume that \( (Z_1, ..., Z_k) \) for \( k < n \) is a particular solution of our problem. The upper bound on this particular solution can be given as

\[ \left( \prod_{j=1}^{k} R_j \right) \left( \prod_{i=k+1}^{n} \max R_{ij} \right). \]  

(4)

The left part (4) reflects the actual contribution of the particular solution to the value of the objective function, while the right part indicates the optimal selection, ignoring cost constraints (i.e. the selection \( (Z_{k+1}^o, ..., Z_n^o) \)). Obviously, \( (Z_1, ..., Z_k) \) cannot provide any possible feasible solution if

\[ \sum_{i=1}^{k} C_i + \sum_{i=k+1}^{n} (\min_{i} C_{ij}) > B. \]

(5)

The upper bound (4) is reached if

\[ \sum_{i=1}^{k} C_i + \sum_{i=k+1}^{n} C_i^o \leq B, \]  

where \( C_i^o \) is the cost, corresponding to \( Z_i^o \) for \( i = k+1, ..., n \).

The proposed procedure for the scheme of branches and borders begins with the selection of a known feasible solution for the problem, which can serve as a lower boundary.

The SR model forms the optimal composition of modules of one functional multiversion GCCCS software system with redundancy, when the multiversion GCCCS software performs one, but the most important (critical in fault tolerance) function. Failure of the software and, therefore, this function of the control and command complex can lead to very serious negative consequences. In this model, software fault tolerance is achieved by introducing redundant versions of each module into the system. Accordingly, restrictions on the cost of software performing such functions should be sufficient to allow module redundancy.

The purpose of the SR model is to determine the optimal composition of modules regarding the redundancy of versions, maximizing the reliability of the multiversion GCCCS software, without violating the framework of cost restrictions.

The target function:

\[ \max R = \max \left\{ \prod_{j=1}^{n} \left( 1 - \prod_{j=1}^{m_i} (1 - R_{ij})^{Z_{ij}} \right) \right\}, \]

According to the restrictions

\[ Z_{ij} = 0, 1; i = 1, ..., n; j = 1, ..., m, \]

\[ \sum_{j=1}^{m_i} Z_{ij} \geq 2, i = 1, n, \sum_{j=1}^{m_i} \sum_{j=1}^{m_i} Z_{ij} C_{ij} \leq B. \]

where
The reliability of the $i$-th module is defined as the probability that at least one of the $m_i$ versions is executed correctly, and the set of restrictions ensures that at least one version is selected for each $i$-th module.

One solution to this problem is in the use of a dynamic programming algorithm. According to the proposed algorithm, we define the state of the software system $S$ as the allowable cost and state for reflecting the $i$-th module ($i = 1, \ldots, n$). Then $R_i(S)$ is the probability of failure-free operation of multiversion software built of the modules $i$, $i = 1, \ldots, n$. By setting $S$ as an allowable cost, we have ($B - S$) thus, the possible costs allowed for modules is $1, \ldots, i - 1$.

$R_i(S) - \max\{[1 - \prod_{j=1}^{m_i} (1 - R_{ij})] \cdot R_{i+1}(B - \sum_{j=1}^{m_i} C_{ij}Z_{ij})\}$

where the maximization follows the values $Z_{ij}$, for which

$\sum_{j=1}^{m_i} Z_{ij} \geq 2$,

$\sum_{j=1}^{m_i} C_{ij}Z_{ij} \leq S$.

The recursive formula for $R_n(S)$ is

$R_n(S) = \max\{[1 - \prod_{j=1}^{m_n} (1 - R_{nj})]\}$

where the maximization follows the values $Z_{nj}$, which require

$\sum_{j=1}^{m_n} Z_{nj} \geq 2$,

$\sum_{j=1}^{m_n} C_{nj}Z_{nj} \leq S$.

According to the set $i$ position, $R_i(S)$ must be determined for all $S$ in the following sequence

$\left[\sum_{k=i}^{n} \min C_{kj}, \ldots, B, \sum_{k=i}^{i-1} \min C_{kj}\right]$.

The MNR model forms the optimal composition of the multifunctional ($K$ functions) multiversion software modules of the GCCCS without redundancy.

The software system consists of several programs, each of which performs its function. Each program contains a number of modules. Programs can be determined by the corresponding functions of the software system, and modules by any program [14]. The goal of this model is to determine the optimal set of modules for programs without using redundancy in such a way that the reliability of the multiversion software system is at its maximum under given cost constraints.

Let $S_k$ denote the set of modules corresponding to the program $k$. For each $i \in S_k$ module there are $m_i$ available versions. For each $i$ Note that the same module can be called by different programs. Let us number all the called modules by numbers from 1 to $n$. The problem can be formulated as:
\[
\text{max } R = \max \left\{ \sum_{k=1}^{K} F_k \prod_{i \in S_k} \sum_{j=1}^{m_i} Z_{ij} R_{ij} \right\}
\]

with restrictions:

\[
\sum_{j=1}^{m_i} Z_{ij} = 1, \quad i = 1, n; \\
\sum_{i=1}^{n} \sum_{j=1}^{m_i} Z_{ij} C_{ij} \leq B; \\
Z_{ij} = 0, 1; i = 1, \ldots, n; j = 1, \ldots, m_i.
\]

The obtained formulation of the problem can be solved by a method similar to that used for the problem with a minor correction. The MR model forms the optimal composition of the multifunctional (\(K\) function) multiversion software modules of the GCCCS with the introduction of redundancy.

Since redundancy is allowed in the structure, it is possible to select more than one version for each module of the software system. We formulate the problem as follows:

\[
\text{max } R = \max \left\{ \sum_{k=1}^{K} F_k \prod_{i \in S_k} \left( 1 - \prod_{j=1}^{m_i} (1 - R_{ij}) Z_{ij} \right) \right\}
\]

with restrictions:

\[
\sum_{j=1}^{m_i} Z_{ij} \geq 2, \quad i = 1, n; \\
\sum_{i=1}^{n} \sum_{j=1}^{m_i} Z_{ij} C_{ij} \leq B; \\
Z_{ij} = 0, 1; i = 1, \ldots, n; j = 1, \ldots, m_i.
\]

Due to the restriction (and, since we are dealing with \(K > 1\) software functions), none of the methods proposed earlier can be directly used to solve this problem. In addition, the objective function is nonlinear, but this problem can be solved using integer programming [14].

For practical applications (large dimensions of statements, algorithmically defined constraints, etc.), when the use of traditional methods does not allow to obtain an acceptable result in real time, it is proposed to apply new modifications of random search algorithms (directional random search), which allow to avoid these disadvantages initially considered solution schemes.

Note that the need to control the correct execution of the software function requires the additional use of COTS support components. However, this is currently an integral part of the highly reliable software systems being created; and for the multiversion software of the GCCCS, the statistical independence of various versions of software modules is guaranteed.

**Conclusion**

The proposed basic models for the formation of multiversion software for control and information processing systems make it possible to implement various fundamentally possible structures of multiversion one- and multifunctional software with redundancy and without redundancy, taking into account its reliability and costs.

The iterative algorithm for dropping out versions of modules on the restrictions on attributes
calculated additively allows us to form the composition of the principally implemented multiversion software scheme for control and information processing systems and, in particular, the GCCCS.

**Acknowledgements**
This research was supported by the Ministry of Education and Science of the Russian Federation (agreement from 26.09.2017 № 14.577.21.0246; unique ID project RFMEFI57717X0246).

**References**

[1] Kotenok A V 2009 *A multiversion environment for faultless software complexes of control systems* (Krasnoyarsk)

[2] Kovalev I V 1997 *Multiversion systems for forming software complexes in satellite control systems* (Krasnoyarsk: Krasnoyarsk State Technical University)

[3] Semen’ko T I and Tsarev R Iu 2005 *Criteria and restrictions of solving problems relevant to the synthesis of optimal modular control and command systems* Molodezh i nauka: nachalo XXI veka (Moscow) 3 pp 28–30

[4] Stentsel’ A V 2008 *Instrumental means of forming the multiversion architecture of faultless software systems* (Krasnoyarsk: Siberian State Aerospace University)

[5] Semen’ko T I and Litiaikina A S 2005 *A discriminate method for multi-attribute problem-solving for the architecture of information systems* Molodezh i nauka: nachalo XXI veka – (Moscow) 3 pp 133–6

[6] Tsarev R Iu 2010 *The formation of reliable software for real-time telecommunications systems* Sovremennye naukoemkie tehnologii 9 119–22

[7] Antamoshkin A N and Kovalev I V 2000 Determining the optimal structure of multiversion software under time and cost restrictions Vestnik of SAA 1 111–24

[8] Ezhmanskaia S N 2005 *The reliability of modular structures in N-variant software systems* Vestnik universitetskogo kompleksa 4(18) 169–74

[9] Kovalev I V, Popov A A and Privalov A S 2000 *An optimal method for designing multiversion control systems* Informatsionnye tehnologii v innovatsionnykh proektakh (Izhevsk) pp 24–9

[10] Kovalev I, Privalov A and Shipovalov Ju 2000 Optimization Reliability Model for TelecommunicationsSoftware Systems, Modelling, Measurement and Control 4–5 47–52

[11] Volkov B A 2007 *The issue of selection a version of multiversion software modules* Vestnik universitetskogo kompleksa 9(23) 68–72

[12] Seme’ko T I 2005 *Optimizing control systems according to the reliability criterion by introducing software and hardware redundancy* Vestnik universitetskogo kompleksa 4(18) 38–43

[13] Tsarev R Iu 2007 *Optimization models of forming multiversion software for faultless control systems* Pribory i sistemy. Upravljenie, kontrol’, diagnostika 9 18–20

[14] Tsarev R Iu and Seme’ko T I 2005 *Forming an optimal content of a multifunctional and a multiversion software system with redundancy* Uspekhi sovremennoi estestvoznaniiia 6 33-4