

УДК 004.021

Doi: 10.31772/2712-8970-2023-24-3-436-449

Для цитирования: Ефремова С. В. Метод мультиверсионного программирования для обработки телеметрической информации малых космических аппаратов // Сибирский аэрокосмический журнал. 2023. Т. 24, № 3. С. 436–449. Doi: 10.31772/2712-8970-2023-24-3-436-449.

For citation: Efremova S. V. [N-version programming for nanosatellite telemetry processing]. *Siberian Aerospace Journal*. 2023, Vol. 24, No. 3, P. 436–449. Doi: 10.31772/2712-8970-2023-24-3-436-449.

Метод мультиверсионного программирования для обработки телеметрической информации малых космических аппаратов

С. В. Ефремова

Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева
Российская Федерация, 660037, г. Красноярск, просп. им. газ. «Красноярский Рабочий», 31
E-mail: efremova_svet@sibsau.ru

Программное обеспечение является ключевым элементом, обеспечивающим функционирование любой современной сложной технической системы. Одной из таких систем являются группировки космических аппаратов и связанные с ними комплексы наземного управления, обеспечивающие прием, передачу и обработку телеметрической информации (ТМИ). Сбор и обработка информации в системах телеметрии обеспечивают процесс управления как самим космическим аппаратом (КА), так и установленным на нем научным оборудованием. При этом телеметрические данные, принимаемые наземными комплексами управления (НКУ) представляют собой огромные объемы данных, обработка которых является сложной и трудоемкой задачей. Для решения этой проблемы используются различные методы автоматической обработки данных. Их совершенствование является ключевым фактором обеспечения отказоустойчивости бортового программно-аппаратного комплекса и повышения его надежности.

Среди существующих методов обработки информации, нашедших широкое применение в исследуемой области, можно выделить метод мультиверсионного программирования (МВП).

Мультиверсионное программирование прочно закрепилось как эффективный метод повышения надежности программного обеспечения и создания отказоустойчивых систем. С момента своего возникновения в 1970-е гг., данный подход также ассоциируется с надежностью программных систем для аэрокосмической отрасли, в том числе наземных пунктов управления космическими аппаратами. В настоящей работе рассматривается применение данного подхода для обработки телеметрической информации, поступающей с малых космических аппаратов. Автором рассмотрен вопрос критики МВП подхода в научной литературе в части его применимости для задач обработки ТМИ.

Ключевые слова: мультиверсионное (N-версионное) программирование, программная избыточность, малый космический аппарат, телеметрическая информация, надежность программного обеспечения.

N-version programming for nanosatellite telemetry processing

S. V. Efremova

Reshetnev Siberian State University of Science and Technology
31, Krasnoyarskii Rabochii Prospekt, Krasnoyarsk, 660037, Russian Federation
E-mail: efremova_svet@sibsau.ru

Software is a key element that ensures the functioning of any modern complex technical system. One such system is the constellation of spacecraft and associated ground control complexes that provide reception, transmission, and processing of collected telemetry. The process of data acquisition and its subsequent processing is critical to the flight control of the spacecraft and its onboard scientific equipment. Furthermore, telemetry data processed by ground control systems involves large volumes of raw data, the processing of which is a complicated and time-consuming task. In order to solve this problem, various methods of automatic data processing are used. Improving them is a key factor in ensuring the fault tolerance of onboard software and hardware, improving its reliability.

Of all the existing widely-used methods of data processing, we shall focus on N-version programming (NVP) approach.

N-version programming has firmly established itself as an effective method for increasing software reliability and designing fault-tolerant systems. Since its inception in the 1970s, this approach has been deeply connected with the development of aerospace software systems, including, among others, satellite ground control stations. In light of the aforementioned, this paper discusses the application of NVP for processing telemetry data gathered from nanosatellites (CubeSats). Due to the fact that there exists a skeptical view on the NVP approach in terms of its efficiency, the author covers this issue in existing literature in terms of the approach's applicability for processing satellite telemetry.

Keywords: multiversion (N-version) programming, software redundancy, nanosatellite (CubeSat), telemetry, software reliability.

Введение

Ключевой технической задачей в области создания программного обеспечения (ПО) является повышение его надежности в части отказоустойчивости. Среди существующих методов, построенных на основе принципа избыточности программных компонент, мультиверсионный подход (МВП) занимает особое место, главным образом, в силу периодического возникновения и угасания интереса к нему в научной литературе. Мультиверсионный подход, известный также как *N*-версионный, был впервые предложен для решения задачи повышения надежности ПО в начале 70-х гг. XX в. Отметим, что рядом зарубежных исследователей *N*-версионное программирование рассматривается лишь как один из вариантов МВП [1], в то время как в большинстве работ данные определения синонимичны. В последующие два десятилетия данный метод получил теоретическое развитие, главным образом, в работах группы американских исследователей во главе с А. Авиженисом и его аспирантами в Университете штата Калифорния [2]. Были сформированы основные фундаментальные принципы данного подхода, отображенные в выражении $N \geq 2$, означающем, что если количество версий программы более или равно двум, имеет место мультиверсионность системы. Несмотря на обилие публикаций А. Авижениса и его коллег в период 1970-х – начала 2000-х гг., а также излишне сложный, на наш взгляд, язык данных работ, базовые принципы МВП являются простыми. Предполагается, что независимыми группами разработчиков создаются две или более версий одной и той же программы в рамках заданных спецификаций. Прогон данных программ (мультиверсий) помогает выявить закономерности программных ошибок и сбоев, позволяя тем самым отобрать оптимальную версию программы. В рамках «классического» МВП подхода выделяются три основных элемента:

1. Процесс первоначальной спецификации и *N*-версионное программирование призваны обеспечить независимость и функциональную эквивалентность *N* числа самостоятельных попыток разработки ПО.

2. Конечный продукт (программа), созданный в рамках МВП подхода и имеющий атрибуты параллельного выполнения с определенными перекрестными точками и векторами сравнения для принятия решений.

3. Среда, обеспечивающая реализацию МВП программы, обеспечивает алгоритмы принятия решений в заданных перекрестных точках [2].

Вводится понятие диверсификации или разнообразия проектирования ПО, однако подчеркивается отличие многоканального, избыточного, ПО от много(мульти)версионных методов.

Идея применения системы с n параллельными каналами и алгоритма голосования предполагает, что независимые отказы происходят в рамках отдельных версий или модулей ПО и не оказывают влияния на систему в целом.

Принцип использования n параллельных каналов с алгоритмом голосования является традиционным методом для повышения надежности как аппаратного, так и программного обеспечения технической системы. В этом случае обоснование МВП подхода с внедренными в него n версиями программ (программных модулей) заключается в следующих постулатах [3]:

- каналы остаются независимыми друг от друга во всех случаях;
- программные сбои всегда приводят к разногласию между дублирующими каналами;
- если алгоритм голосования правильно функционирует, вероятность того, что не менее двух каналов ($N \geq 2$) из общего числа n соответствует

$$p(m \geq 2) = 1 - p(m = 1) - p(m = 0),$$

где m – количество согласующих каналов; p – вероятность отказа любого из них при входном сигнале. Вероятность системной ошибки в данном случае будет

$$1 - p(m \geq 2) = p(m = 0) + p(m = 1) = p^n + n(1 - p)p^{n-1}.$$

Таким образом, использование мультиверсионной диверсификации ведет к определенному улучшению в рамках одного программного канала:

$$\frac{p}{p^n + n(1 - p)p^{n-1}} = \frac{1}{p^{n-1} + n(1 - p)p^{n-2}}.$$

В последнее десятилетие МВП подход получил развитие в работах ряда отечественных и зарубежных исследователей [4–18]. Так, в работе [18] рассмотрено применение данного подхода для создания отказоустойчивого ПО для динамических систем (беспилотных летательных аппаратов). Отметим, что применение МВП в проектировании авиационного ПО заявлялось на ранних этапах в качестве основной сферы успешного применения данного метода [1; 18]. Однако данное использование МВП, в некотором смысле, противоречит классической концепции МВП (описанной А. Авиженисом в его «Методологии» [2]), которая строится вокруг процесса проектирования ПО, а не его последующей эксплуатации.

Проблема использования МВП подхода

После своего возникновения, МВП подход долгое время не подвергался критике со стороны научного сообщества, главным образом, по причине в основном его теоретического значения.

Несмотря на обилие научной литературы на тему применения МВП для обработки различных категорий и типов данных, методика реализации этого метода крайне скупо освещена. В этой связи выгодно отличается вышеупомянутое исследование [19–21]. Авторы работы подробно прописывают весь алгоритм организации эксперимента по испытанию МВП метода для анализа данных, полученных от заданных в рамках эксперимента условий. Данное исследование является для нас особенно важным, поскольку по результатам его авторы пришли к выводам о неэффективности МВП (хотя они указывают, что этот вывод справедлив только в рамках проведенного ими эксперимента). Нами было принято решение повторить данный эксперимент с изменением параметров в части типа данных и их объема.

Обратимся к описанию эксперимента.

Применяемые в рамках эксперимента программы, по условиям, были написаны на языке программирования Pascal. В связи с тем, что данный язык можно считать устаревшим, нами было принято решение о написании программ на более современном языке Python (или C++). Авторами эксперимента были предложены 27 версий одной и той же программы, написанной независимыми друг от друга коллективами программистов. Одна версия – 28-я – была определена, как «эталонная» и применялась для калибровки остальных версий. В данной работе число версий программы было доведено до 50 (см. табл.).

Данные о сбоях мультиверсий по результатам эксперимента Найта и Левисона

Версия	Сбой	Надежность	Версия	Сбой	Надежность
1	2	0,999998	15	0	1,000000
2	0	1,000000	16	62	0,999938
3	2297	0,997703	17	269	0,999731
4	0	1,000000	18	115	0,999885
5	0	1,000000	19	264	0,999736
6	1149	0,998851	20	936	0,999064
7	71	0,999929	21	92	0,999908
8	323	0,999677	22	9656	0,990344
9	53	0,999947	23	80	0,999920
10	0	1,000000	24	260	0,999740
11	554	0,999446	25	97	0,999903
12	427	0,999573	26	883	0,999117
13	4	0,999996	27	0	1,000000

Существуют два варианта подхода к формированию версий предложенной программы:

- непосредственное написание программ вручную, как это было сделано в [19];
- моделирование версий ПО на основе математических моделей.

Поскольку при конкретных заданных параметрах, версии ПО, предположительно, будут во многом схожи, если не идентичны, привлечение человеческих ресурсов для их создания не может быть признано целесообразным. В этой связи оптимальным решением следует признать использование математической модели для формирования последующих версий заданного ПО. Таким образом, нами была создана «эталонная» версия программы, которая затем была воспроизведена 49 раз посредством, представленных ниже математических методов.

Для реализации задач настоящей статьи рассмотрим МВП модель на примере МВП структурной подмодели. Данный подход показан впервые в работе [22] и представляет особый интерес, поскольку включает в себя базовые принципы МВП подхода в рамках теории надежности ПО. В рамках данного метода можно объединить функциональные и временные сбои в единое значение. Это позволяет построить аналитическую модель на основе как функциональных, так и производительных сбоев. В подмодели МВП время рассматривается в качестве постоянной величины и измеряется с момента прогона мультиверсий программы. Для данной модели выдвинуты следующие предположения:

- версии ПО условно независимы друг от друга в заданном вводе;
- время сбоя программных версий для заданного ввода представляет собой одинаково распределенные произвольные переменные с плотностью вероятности $f_F(t; \mathbf{v})$ в зависимости от d размерного вектора параметра $\mathbf{v} = (v_1, \dots, v_d)$;
- время выполнения программных версий для данного ввода представляет равномерно распределенные переменные с плотностью вероятности $f_E(t; \boldsymbol{\psi})$ в зависимости от b размерного вектора параметра: $\boldsymbol{\psi} = (\psi_1, \dots, \psi_b)$;
- время выполнения алгоритма голосования незначительно в сравнение с требуемым временем для реализации каждой версии;
- в связи с реальным ограничением по времени, система должна выполнять правильные решения во временном интервале $\tau > 0$.

Далее приведем модифицированную версию реализации МВП в заданной подсистеме на основе [22]. Функция распределения $F_F(t; \mathbf{v})$ дает вероятность отказа первой версии программы до t с учетом отказа функциональности. В таком случае вероятность того, что первая версия имеет функциональный отказ составляет

$$P_f(\tau; \mathbf{v}, \boldsymbol{\psi}) = \int_0^{\tau} F_F(t; \mathbf{v}) f_E(t; \boldsymbol{\psi}) dt. \quad (1)$$

Далее, предположим, что каждая версия выдает правильный результат до того, как τ :

$$P_e(\tau; v, \psi) = \int_0^{\tau} [1 - F_F(t; v)] f_E(t; \psi) dt. \quad (2)$$

В этом случае отказ производительности системы, а также временной отказ, возникают, если ни одна из версий не выполнена до заданного времени τ :

$$P_{ne}(\tau; \psi) = 1 - \int_0^{\tau} f_E(t; \psi) dt = 1 - F_E(\tau; \psi). \quad (3)$$

Далее рассмотрим модель с мультиверсиями программы. Временной отказ МВП системы составляет $(n = 2m - 1)$ для заданного ввода в случае, если большинство версий не выводят обработанные данные (заканчивают свою операцию) во время $\leq \tau$:

$$P_{ff}(\tau; \psi) = \sum_{n_3=m}^n \binom{n}{n_3} P_{ne}^{n_3}(\tau; \psi) [1 - P_{ne}(\tau; \psi)]^{n-n_3}. \quad (4)$$

В случае если большинство версий своевременно завершили работу (до τ), существует высокая вероятность функционального сбоя всей системы (большинство результатов ошибочны):

$$P_{ff}(\tau; v, \psi) = \sum_{n_2=m}^n \binom{n}{n_2} P_f^{n_2}(\tau; v, \psi) [1 - P_f(\tau; v, \psi)]^{n-n_2}. \quad (5)$$

Напротив, большинство результатов верны (мультиверсии прогнаны успешно):

$$P_{ok}(\tau; v, \psi) = \sum_{n_1=m}^n \binom{n}{n_1} P_e^{n_1}(\tau; v, \psi) [1 - P_e(\tau; v, \psi)]^{n-n_1}. \quad (6)$$

Наконец в уравнении (7) показано отсутствие как большинства положительных, так и большинства отрицательных результатов:

$$P_{nm}(\tau; v, \psi) = 1 - P_{ff}(\tau; \psi) - P_{ok}(\tau; v, \psi) - P_{ff}(\tau; v, \psi). \quad (7)$$

Обработка телеметрической информации с МКА

ТМИ, передаваемая на наземный комплекс управления (НКУ), может быть в различных форматах, включая тексты, изображения, аудио и видеофайлы.

Телеметрические системы состоят из следующих элементов:

1. Система сбора данных.
2. Одна из следующих мультиплексных систем:
 - разделяемые по частоте (частотное мультиплексирование);
 - разделяемые по времени (дискретные, временное мультиплексирование);
 - гибридные системы, представляющие собой сочетание систем, разделяемых по частоте и по времени.
3. Модулятор, передатчик, антенна.
4. Волнообразующий и передающий канал связи.
5. Антенна, приемник радиочастот, секция промежуточной частоты, демодулятор сигнала.
6. Демультимплексная система для частотных и временных систем, а также их гибридов.
7. Система обработки данных [23].

Первые шесть представленных элементов отвечают за сбор различных физических данных, их преобразование в электронный сигнал с последующим преобразованием в различные частоты с учетом дискретизации с целью их передачи. Частоты сигнала передачи обычно находятся в пределах двух диапазонов: 1435–1535 МГц и 2200–2290 МГц. Не останавливаясь подробно на системе волнообразования, рассмотрим пятый, шестой и седьмой элементы представленной

системы. Они состоят из аппаратного обеспечения, отвечающего за прием сигналов с КА, а также аппаратного и программного обеспечения, осуществляющих последующую обработку данных и их преобразование в проектные форматы. Демультимплексная подсистема обеспечивает разделение частотных и дискретных сигналов и их направление от отдельных датчиков в правильные каналы, после чего данные могут быть отображены, записаны и подвергнуты дальнейшей обработке.

Обратимся к проблемам, возникающим на этапе обработки телеметрической информации. Ввиду характера своей деятельности, КА должны обеспечивать в максимально кратчайшем временном промежутке компактные, неискаженные и доступные библиотеки данных. В этой связи одной из основных проблем является ограниченная полоса пропускания, доступная для передачи телеметрических данных. Доступная полоса пропускания ограничена пропускной способностью системы связи и расстоянием между космическим аппаратом и наземной станцией. Эта ограниченная полоса пропускания создает проблему для обработки и анализа телеметрических данных в режиме реального времени.

Еще одной проблемой является сложность архитектуры телеметрических данных. Телеметрические данные с космического аппарата обычно состоят из большого количества параметров, каждый из которых имеет свой диапазон значений и единиц измерения. Анализ этих данных требует специальных знаний и опыта, которые не всегда могут быть доступны.

Наконец, существует угроза потери или повреждения данных во время передачи. В некоторых случаях телеметрические данные могут быть потеряны или повреждены во время передачи из-за помех или других факторов. Это может привести к неполным или неточным данным, что может повлиять на процесс анализа и принятия решений.

Обработка данных телеметрии космических аппаратов является одной из наиболее сложных задач в области обработки космических данных. Форматы данных телеметрии космических аппаратов сложны и разнообразны, а определения форматов данных для разных платформ космических аппаратов различаются. Общие форматы данных включают формат кадра РСМ, формат пакета, формат смешанного кадра, формат кадра с подсчетом циклов и так далее. С появлением новой платформы количество форматов постоянно увеличивается. Форматы данных телеметрии космических аппаратов имеют ряд сложных характеристик: форматы имеют иерархическую и вложенную структуры, которые необходимо обрабатывать перекрестными кадрами; форматы имеют сложные зависимости параметров. Модель обработки данных телеметрии КА существующего программного обеспечения обработки данных миссии основана на структуре «Кадр – Поле». Различные форматы кадров описываются различными методами обработки кадров телеметрии, и каждое поле в формате кадра описывает формат конкретного параметра. Данная модель имеет следующие проблемы:

1. Описательная способность метода обработки одного кадра телеметрии ограничена, что не позволяет адаптироваться к характеристике: форматы имеют иерархическую и вложенную структуру, которые необходимо обрабатывать перекрестными кадрами.
2. Сложные зависимости между параметрами не могут быть эффективно описаны.
3. Универсальность и масштабируемость модели плохие. Новое изменение (даже небольшое изменение) формата данных приводит к перестройке метода обработки кадров, что означает частые изменения программного кода.

Таким образом, существующую модель обработки данных телеметрии КА, основанную на структуре «Кадр – Поле», трудно адаптировать к реальной ситуации частых изменений форматов данных телеметрии КА, особенно в период высокочастотных полетов. Необходимо разработать новую модель обработки данных телеметрии КА, обладающую большей выразительностью, большей универсальностью и масштабируемостью и решающую вышеперечисленные проблемы [24].

К ключевым отличиям телеметрических данных, полученных с МКА, в отличие от штатных КА, следует отнести условия их генерации: так, крупные аппараты способны размещать в себе

большее количество компонентов, обеспечивающих избыточность системы, тем самым повышая ее отказоустойчивость в условиях агрессивной космической среды (к частности, ионизирующего излучения) [25].

К ранним образцам (первому поколению) специализированного ПО для МКА относится программная архитектура PolySat для аппаратов серии СР. Платформа аппаратного обеспечения с повышенной отказоустойчивостью была построена, главным образом, с применением избыточных компонентов; их относительная низкая стоимость и малое энергопотребление позволили выстроить систему с высокой степенью избыточности. На рис. 1 представлено устройство аппаратного обеспечения КА.

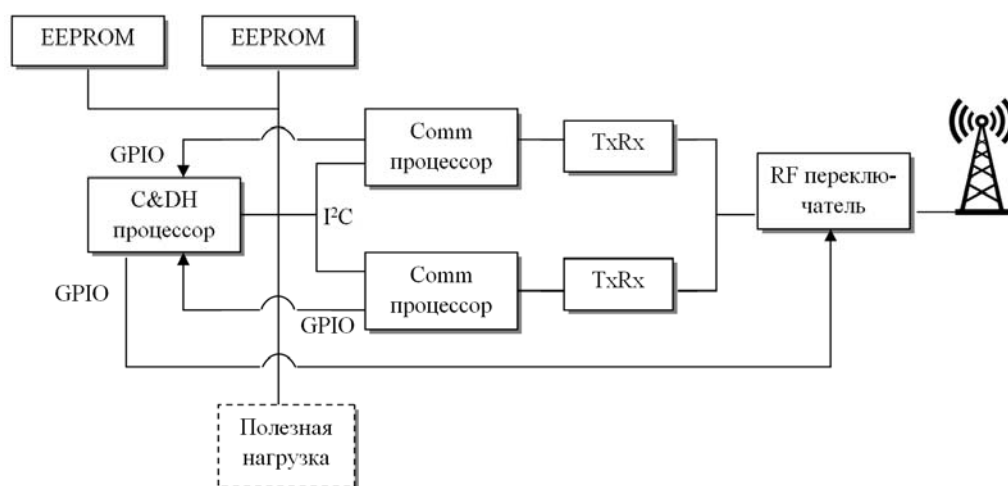


Рис. 1. Пример аппаратной блок-схемы резервирования коммуникационной системы

Fig. 1. Example of hardware block diagram of redundant communication system

Проектируемый КА работал в трех режимах: подготовительном (pre-ops), нормальном (normal-ops) и аварийном (contingency). При этом выбор режима для связи и контроллера команды и обработки данных (C&DH) осуществляется независимо. Последний отвечает как за различные аспекты работы системы, так и за сбор, обработку и передачу телеметрии [25–27]. Данные, собранные с трех аппаратов в спутниковой группировке, собирались и хранились в I²C последовательной асимметричной шине электрически стираемого перепрограммируемого ПЗУ. Объем памяти составлял не более 256 кб, что существенно ограничивало эксплуатационные характеристики МКА. Второе поколение ПО построено на Linux; так, автор [25] предлагает свой вариант программного пакета для приема и обработки телеметрии.

В нашем случае, ТМИ, принятая с ReshU-1, сохраняется в файлах журнала (лог-файлах; расширение CSV) со всеми фреймами (кадрами) данных (рис. 2–3). Причем, фреймы для каждой команды отличаются. Разбор данных осуществляется синтаксическим анализатором (парсером) после чего они сохраняются в базе данных лаборатории. Лог-файлы являются распространенным форматом для такого вида информации, как ТМИ. Современные измерительные информационные системы способны генерировать постоянные потоки данного формата файлов, дающих сведения о работе и состоянии системы.

Интеграция мультиверсионного подхода в процесс обработки телеметрии

Одним из важных шагов является интеграция мультиверсионного подхода в систему обработки телеметрии. Это предполагает создание структуры, которая может обрабатывать несколько версий алгоритма и определять конечный результат на основе консенсуса версий. Система также должна быть разработана для обработки любых ошибок или несоответствий, которые могут возникнуть в процессе обработки телеметрических данных.

```
{
  radio: {
    name: "gr-satnogs",
    version: "v2.3-compat-xxx-v2.3.4.0",
    parameters: {
      soapy-rx-device: "driver=rtlsdr",
      samp-rate-rx: "2.048e6",
      rx-freq: "435380000",
      file-path: "/tmp/.satnogs/data/receiving_satnogs_8101204_2023-08-31T22-36-08.out",
      waterfall-file-path: "/tmp/.satnogs/data/receiving_waterfall_8101204_2023-08-31T22-36-08.",
      decoded-data-file-path: "/tmp/.satnogs/data/data_8101204",
      doppler-correction-per-sec: null,
      lo-offset: null,
      ppm: null,
      rigctl-port: "4532",
      gain-mode: "Overall",
      gain: "22.9",
      antenna: "RX",
      dev-args: null,
      stream-args: null,
      tune-args: null,
      other-settings: null,
      dc-removal: null,
      bb-freq: null,
      bw: null,
      enable-iq-dump: "0",
      iq-file-path: null,
      udp-dump-host: null,
      udp-dump-port: 57356,
      wpm: null,
      baudrate: "19200",
      framing: "ax25"
    }
  },
  latitude: 38.048,
  longitude: 23.739,
  elevation: 104,
  frequency: 435380000
}
```

Рис. 2. Пример записи метаданных с МКА ReshU-1

Fig. 2. ReshU-1 CubeSat metadata sample

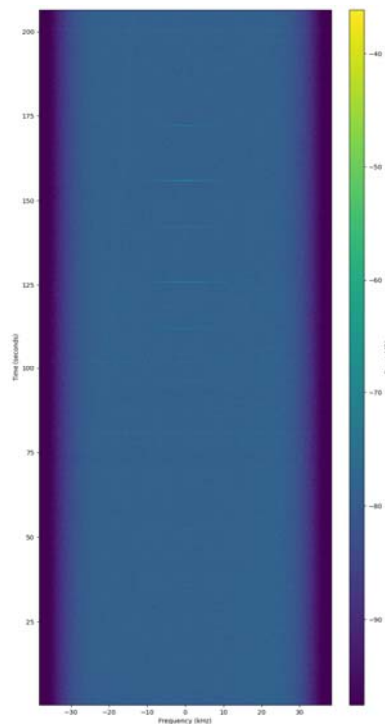


Рис. 3. Участок спектрограммы сигнала телеметрии МКА ReshU-1

Fig. 3. ReshU-1 CubeSat telemetry signal spectrogram

Интеграция мультиверсионного подхода в обработку телеметрии имеет несколько преимуществ. Во-первых, это повышает надежность и точность телеметрических данных за счет снижения вероятности ошибок или несоответствий. Во-вторых, сокращается объем данных, которые необходимо передавать на землю, поскольку передается только окончательный результат, который определяется консенсусом версий. Это может привести к значительной экономии средств и повышению эффективности.

На рис. 4 представлен алгоритм ранжирования телеметрической информации.

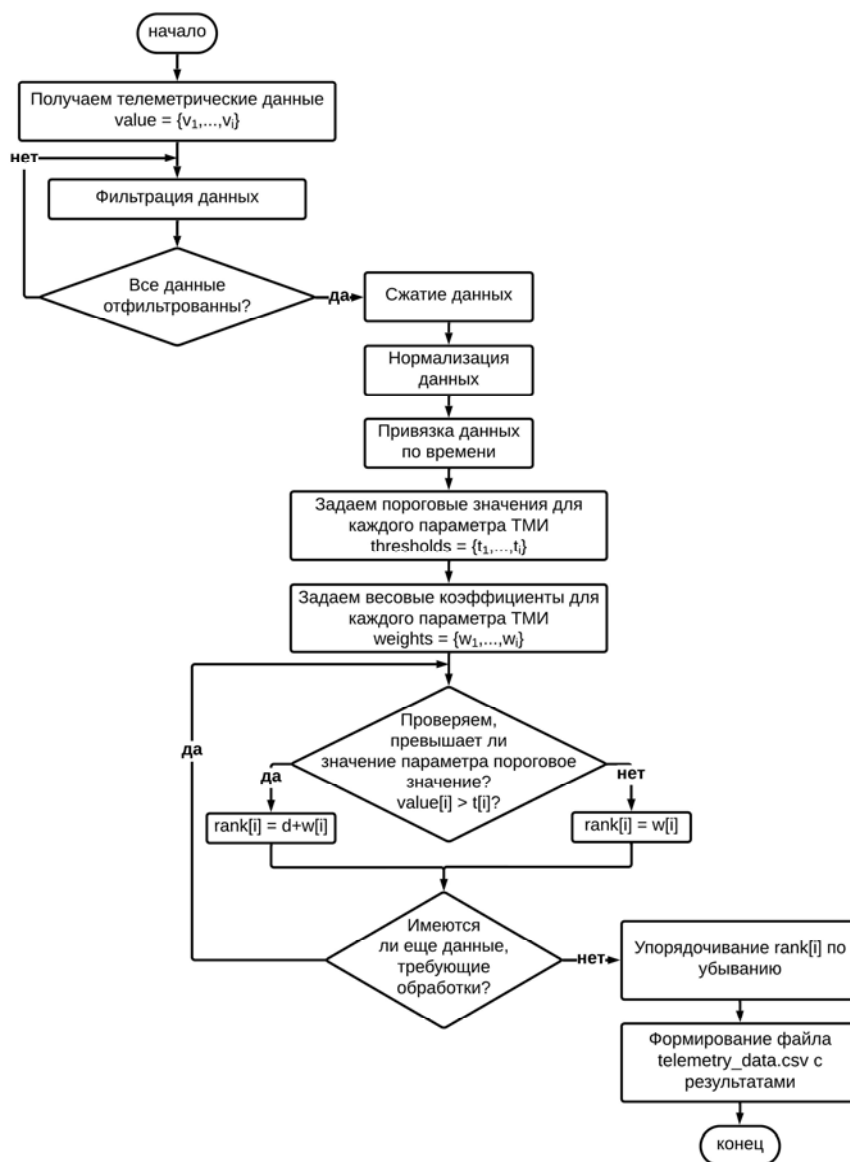


Рис. 4. Алгоритм ранжирования телеметрической информации, поступающей с МКА

Fig. 4. Algorithm for ranking CubeSat telemetry

Процесс ранжирования ТМИ МКА предполагает выполнение следующих шагов:

1. Сбор телеметрических данных с МКА (иных КА).
2. Предварительная обработка данных с целью фильтрации неактуальной или зашумленной информации.
3. Применение алгоритма сжатия для уменьшения размера данных и повышения эффективности передачи.
4. Использование бортовой обработки данных для анализа и извлечения особенностей из данных.

5. Внедрение модели машинного обучения для выявления закономерностей и аномалий в данных.
6. Разработка системы оценок, основанной на приоритетах каждого фрагмента информации.
7. Применение мультиверсионного подхода, разработав несколько версий алгоритма с различными параметрами и конфигурациями (рис. 5).
8. Тестирование каждой версии алгоритма на репрезентативном наборе телеметрических данных.
9. Использование системы оценок для ранжирования производительности каждой версии алгоритма.
10. Выбор алгоритма с наилучшими показателями для каждого набора телеметрических данных.

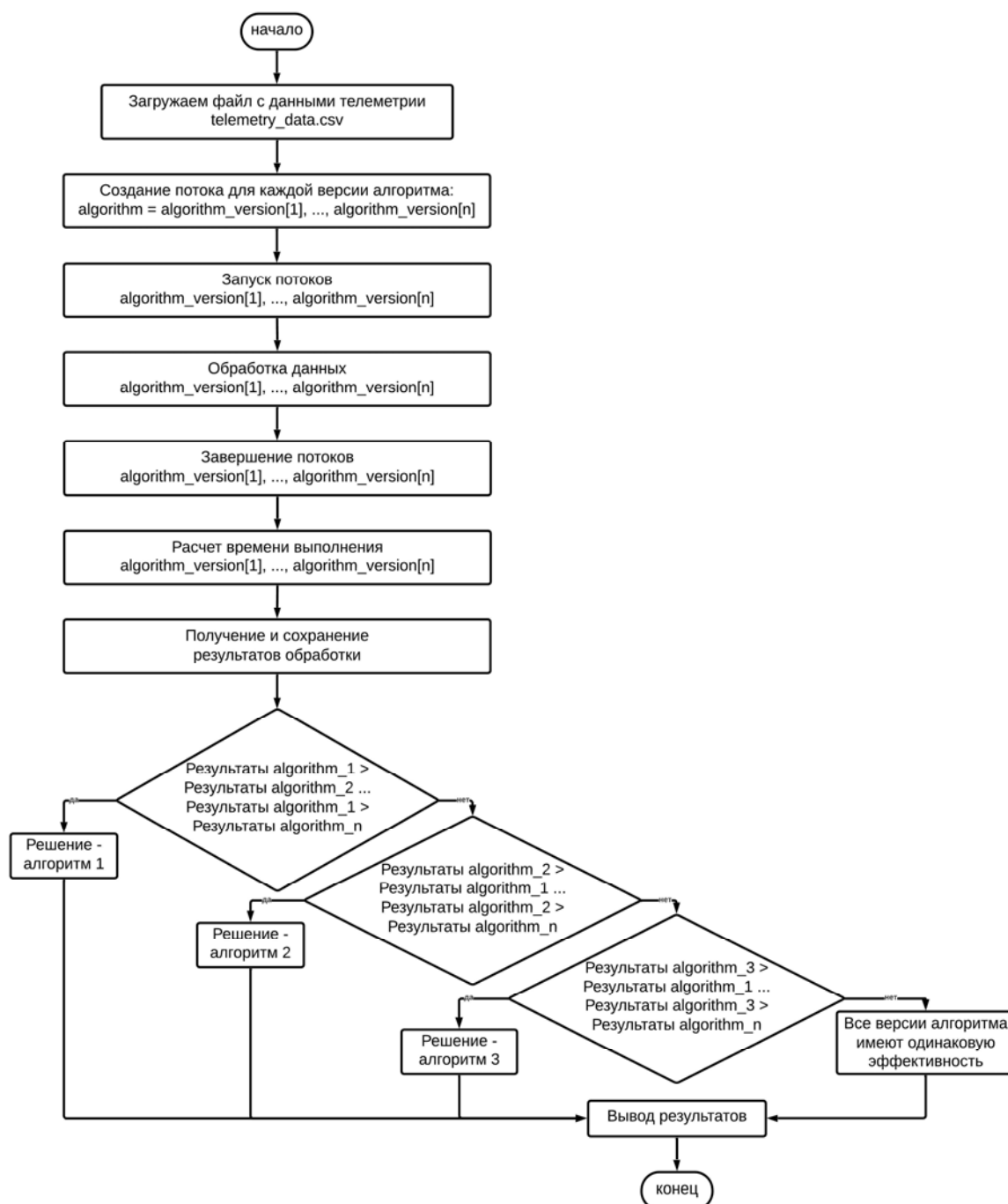


Рис. 5. Введение мультиверсионного подхода в процесс обработки телеметрической информации с МКА

Fig. 5. The N-version approach for processing CubeSat telemetry

Алгоритм ранжирования должен быть гибким и адаптируемым к различным типам телеметрических данных и методам обработки. Он также должен быть способен обрабатывать данные в режиме реального времени и обновлять рейтинг по мере поступления новых данных. Используя МВП подход и систему ранжирования, алгоритм может повысить надежность и точность обработки телеметрической информации с космических аппаратов.

Мультиверсионность ПО для обработки ТМИ является естественным ходом эволюции данных систем. При этом, построение мультиверсий может быть реализовано не только в рамках различных языков программирования, но и различных операционных средах, например Linux. Немаловажным фактором, способствующим продвижению Unix-платформ для достижения поставленных задач в рамках МВП, является их доступность для широкого круга программистов, наличие открытых кодов и бесплатных библиотек.

Анализ ПО, используемого в существующих МКА, показал, что программная архитектура большей части аппаратов построена на принципе многослойности, что позволяет предложить внедрение МВП в компоненты программной архитектуры. В случае с американской МКА платформой KubOS, мультиверсионность уже заложена в концепцию ее архитектуры, которая использует сочетание двух операционных систем Linux и U-boot, выполняющих дублирующие функции, создавая тем самым программную избыточность. Данный подход, который выходит за рамки традиционного МВП подхода, предполагающего лишь создание дублирующих версий, по сути, одной и той же программы, предлагает применение концептуально разного ПО в качестве мультиверсий.

Заключение

Рассмотрены основные проблемы надежностного формирования и характерные особенности программного обеспечения отказоустойчивых систем управления. Приводятся описания причин возникновения программных сбоев и методов обеспечения отказоустойчивости. Показано, что одной из основных задач при разработке программного обеспечения управления МКА является создание таких алгоритмов и методов разработки ПО, которые могут обеспечить устойчивость всей системы к сбоям.

Применяя методологию мультиверсионного формирования программных средств, можно не только обеспечить заданный уровень надежности, но и гарантировать отказоустойчивость систем управления и обработки информации. Данная методология основывается на программной избыточности, введение которой позволяет существенно повысить уровень надежности программной составляющей.

Библиографические ссылки

1. Tröger P. Dependable Systems Software Dependability. 2010.
2. Avizienis A. The methodology of n-version programming // Software Fault Tolerance. 1995. Vol. 3. P. 23–46.
3. Hatton L. N-version design versus one good version // IEEE Software. 1997. Vol. 14, №. 6. P. 71–76.
4. Царев Р. Ю. Среда исполнения мультиверсионного программного обеспечения // Программные продукты и системы. 2007. № 2. С. 29–30.
5. Solving navigation-temporal tasks in different coordinate systems / V. E. Chebotarev, V. V. Brezitskaya, I. V. Kovalev et al. // IOP Conference Series: Materials Science and Engineering. 2018. P. 022029.
6. Development of methods for equivalent transformation of gert networks for application in multi-version software / M. V. Saramud, P. V. Zelenkov, I. V. Kovalev et al. // IOP Conference Series: Materials Science and Engineering. 2016. P. 012015.
7. Applying filtering for determining the angular orientation of spinning objects during interference / I. N. Kartsan, A. E. Goncharov, P. V. Zelenkov et al. // IOP Conference Series: Materials Science and Engineering. 2016. P. 012020.

8. Карцан И. Н., Ефремова С. В. Мультиверсионная модель программного обеспечения систем управления космическим аппаратом с ранжированием принятия решения // Сибирский аэрокосмический журнал. 2021. Т. 22, № 1. С. 32–46.
9. Эффективность радионавигационных систем / И. Н. Карцан, К. Г. Охоткин, Р. В. Карцан, Д. Н. Пахоруков // Вестник СибГУ. 2013. № 3 (49). С. 48–50.
10. Карцан И. Н. Наземный комплекс управления для малых космических аппаратов // Вестник СибГУ. 2009. № 3 (24). С. 89–92.
11. Ковалев И. В., Царев Р. Ю. Многоатрибутивная модель формирования гарантоспособного набора проектов мультиверсионных программных систем // Вестник НИИ СУВПТ. 2001. Вып. 7. С. 129–137.
12. Карцан И. Н., Ефремова С. В., Горовой Д. С. Применение процедуры topsis в интересах оптимизации системы управления // Вопросы контроля хозяйственной деятельности и финансового аудита, национальной безопасности, системного анализа и управления : сб. материалов VI Всеросс. науч.-практ. конф. М., 2021. С. 436–445.
13. Efremova S. V., Kartsan I. N., Zhukov A. O. An ordered ranking multi-attributive model for decision-making systems with attributes of control systems software // IOP Conference Series: Materials Science and Engineering. 2021. P. 12068.
14. The hardware and software implementation of the adaptive platform for an onboard spacecraft control system / I. N. Kartsan, A. O. Zhukov, A. O. Platonov, S. V. Efremova // Journal of Physics: Conference Series. 2019. P. 33071.
15. Formation of optimal composition of the modules of single-function multiversion software for automated control system of the satellite communication system / V. I. Kudymov, V. V. Brezitskaya, P. V. Zelenkov et al. // IOP Conference Series: Materials Science and Engineering, 2018. No. 450 (5). P. 052009.
16. Choice of optimal multiversion software for a small satellite ground-based control and command complex / I. N. Kartsan, S. V. Efremova, V. V. Khrapunova, M. I. Tolstopiatov // IOP Conference Series: Materials Science and Engineering, 2018. No. 450 (2). P. 022015.
17. Карцан И. Н. Мультиверсионное программное обеспечение бортового комплекса управления с генетическим алгоритмом // Решетневские чтения : материалы XXI Междунар. науч.-практ. конф. (08–11 ноября 2017, г. Красноярск). Красноярск : СибГУ им. М.Ф. Решетнева, 2017. Т. 1. С. 372–373.
18. Subasi N., Guner U., Ustolgu I. N-version programming approach with implicit safety guarantee for complex dynamic system stabilization applications // Measurement and Control. 2021. Vol. 54(3–4). P. 269–278.
19. Knight J., Leveson N. A large scale experiment in N-version programming // Proc. Of Ninth Annual Software Engineering Workshop. 1984.
20. Spatial filtering algorithms in adaptive multi-beam hybrid reflector antennas / V. N. Tyapkin, I. N. Kartsan, D. D. Dmitriev, A. E. Goncharov // 2015 International Siberian Conference on Control and Communications, SIBCON 2015 – Proceedings. 2015. P. 7147244.
21. Phase methods for measuring the spatial orientation of objects using satellite navigation equipment / Y. L. Fateev, D. D. Dmitriev, V. N. Tyapkin et al. // IOP Conference Series: Materials Science and Engineering. 2015. P. 012022.
22. Goseva-Popstojanova K., Grnarov A. Performability and reliability modeling of n version fault tolerant software in real time systems // EUROMICRO 97. Proceedings of the 23rd EUROMICRO Conference: New Frontiers of Information Technology (Cat. No. 97TB100167). IEEE, 1997. P. 532–539.
23. Carden F., Jedlicka R., Henry R. Telemetry Systems Engineering. Boston and London: Artech House, 2002.
24. Bin S., Hua W., Yu-jie Y., Hui-fen D., Juan Z. A universal spacecraft telemetry data processing model based on MCP // 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA). Beijing, China, 2017. P. 12–15.

25. Manyak G. Fault tolerant and flexible cubesat software architecture. California Polytechnic State University, 2011.

26. Карцан И. Н., Скрипачев В. О. Оптимизация отказоустойчивого программного обеспечения // Вопросы контроля хозяйственной деятельности и финансового аудита, национальной безопасности, системного анализа и управления: сб. материалов V Всеросс. науч.-практ. конф. 2020. С. 337–341.

27. Карасева М. В., Карцан И. Н., Зеленков П. В. Метопоисковая мультилингвистическая система // Вестник СибГУ. 2007. № 3 (16), С. 69–70.

References

1. Tröger P. Dependable Systems Software Dependability. 2010.
2. Avizienis A. The methodology of n-version programming. *Software Fault Tolerance*. 1995, Vol. 3, P. 23–46.
3. Hatton L. N-version design versus one good version. *IEEE Software*. 1997, Vol. 14, No. 6, P. 71–76.
4. Tsarev R. Iu. [Multiversion software execution environment]. *Programmnye product i sisyemy*. 2007, No 2, P. 29–30 (In Russ.).
5. Chebotarev V. E., Brezitskaya V. V., Kovalev I. V., Kartsan I. N., Malanina Y. N., Shemyakov A. O. Solving navigation-temporal tasks in different coordinate systems. *IOP Conference Series: Materials Science and Engineering*. 2018, P. 022029.
6. Saramud M. V., Zelenkov P. V., Kovalev I. V., Kovalev D. I., Kartsan I. N. Development of methods for equivalent transformation of gert networks for application in multi-version software. *IOP Conference Series: Materials Science and Engineering*. 2016, P. 012015.
7. Kartsan I. N., Goncharov A. E., Zelenkov P. V., Kovalev I. V., Fateev Y. L., Tyapkin V. N., Dmitriev D. D. Applying filtering for determining the angular orientation of spinning objects during interference. *IOP Conference Series: Materials Science and Engineering*. 2016, P. 012020.
8. Kartsan I. N., Efremova S. V. [Multiversion model of software control systems for space vehicles with range of decision-making]. *Siberian Aerospace Journal*. 2021, No 1 (22), P. 32–46 (In Russ.).
9. Kartsan I. N., Okhotkin K. G., Kartsan R. V., Pakhorukov D. N. [Effectiveness of radio-navigation systems]. *Vestnik SibGAU*. 2013, No. 3 (49), P. 48–50 (In Russ.).
10. Kartsan I. N. [Land control complex for small space vehicles]. *Vestnik SibGAU*. 2009, No 3 (24), P. 89–92.
11. Kovalev I. V., Tsarev R. Yu. [A multi-attribute model for building redundant N-version software systems]. *Vestnik NII SUVPT*. 2007, No 7, P. 129–137.
12. Kartsan I. N., Efremova S. V., Gorovoi D. S. [Applying the topsis approach for optimizing control systems] *Sbornik materialov VI Vserossiyskoy nauchno-prakticheskoy konferentsii "Voprosy kontrolya khozyaystvennoy deyatel'nosti i finansovogo audita, natsional'noy bezopasnosti, sistemnogo analiza i upravleniya"* [Collection of materials of the VI All-Russian Scientific and Practical Conference "In the collection: Issues of control of economic activity and financial audit, national security, system analysis and management"]. Moscow, 2021, P. 436–445.
13. Efremova S. V., Kartsan I. N., Zhukov A. O. An ordered ranking multi-attributive model for decision-making systems with attributes of control systems software. *IOP Conference Series: Materials Science and Engineering*. 2021, P. 12068.
14. Kartsan I. N., Zhukov A. O., Platonov A. O., Efremova S. V. The hardware and software implementation of the adaptive platform for an onboard spacecraft control system. *Journal of Physics: Conference Series*. 2019, P. 33071.
15. Kudymov V. I., Brezitskaya V. V., Zelenkov P. V., Kartsan I. N., Malanina Yu. N. Formation of optimal composition of the modules of single-function multiversion software for automated control

system of the satellite communication system. *IOP Conference Series: Materials Science and Engineering*. 2018, No. 450 (5), P. 052009.

16. Kartsan I. N., Efremova S. V., Khrapunova V. V., Tolstopiatov M. I. Choice of optimal multiversion software for a small satellite ground-based control and command complex. *IOP Conference Series: Materials Science and Engineering*. 2018, No. 450 (2), P. 022015.

17. Kartsan I. N. [The multiversion software of the onboard control complex with genetic algorithm] *Materialy XXI Mezhdunar. nauch. konf. "Reshetnevskie chteniya"* [Materials XXI Intern. Scientific. Conf "Reshetnev reading"]. Krasnoyarsk, 2017, P. 372–373 (In Russ.).

18. Subasi N., Guner U., Ustolgu I. N-version programming approach with implicit safety guarantee for complex dynamic system stabilization applications. *Measurement and Control*. 2021, Vol. 54(3–4), P. 269–278.

19. Knight J., Leveson N. A large scale experiment in N-version programming. *Proc. Of Ninth Annual Software Engineering Workshop*. 1984.

20. Tyapkin V. N., Kartsan I. N., Dmitriev D. D., Goncharov A. E. Spatial filtering algorithms in adaptive multi-beam hybrid reflector antennas. *2015 International Siberian Conference on Control and Communications, SIBCON 2015 – Proceedings*. 2015, P. 7147244.

21. Fateev Y. L., Dmitriev D. D., Tyapkin V. N., Kartsan I. N., Goncharov A. E. Phase methods for measuring the spatial orientation of objects using satellite navigation equipment. *IOP Conference Series: Materials Science and Engineering*. 2015, P. 012022.

22. Goseva-Popstojanova K., Grnarov A. Performability and reliability modeling of n version fault tolerant software in real time systems. *EUROMICRO 97. Proceedings of the 23rd EUROMICRO Conference: New Frontiers of Information Technology (Cat. No. 97TB100167)*. IEEE, 1997, P. 532–539.

23. Carden F., Jedlicka R., Henry R. Telemetry Systems Engineering. Boston and London: Artech House, 2002.

24. Bin S., Hua W., Yu-jie Y., Hui-fen D., Juan Z. A universal spacecraft telemetry data processing model based on MCP. *2nd IEEE International Conference on Computational Intelligence and Applications (ICCI)*. Beijing, China, 2017, P. 12–15. DOI: 10.1109/CIAPP.2017.8167051.

25. Manyak G. Fault tolerant and flexible cubesat software architecture. California Polytechnic State University, 2011.

26. Kartsan I. N., Skripachev V. O. Optimizatsiya otkazoustoychivogo programmnogo obespecheniya [Optimizing fault-tolerant software]. *Sbornik materialov V Vserossiyskoy nauchno-prakticheskoy konferentsii "Voprosy kontrolya khozyaystvennoy deyatelnosti i finansovogo audita, natsional'noy bezopasnosti, sistemnogo analiza i upravleniya"* [Collection of materials of the VI All-Russian Scientific and Practical Conference "Issues of control of economic activity and financial audit, national security, system analysis and management"]. Moscow, 2020, P. 337–341.

27. Karaseva M. V., Kartsan I. N., Zelenkov P. V. [Meta-search multi-linguistic system]. *Vestnik SibGAU*. 2007, No. 3 (16), P. 69–70 (In Russ.).

© Ефремова С. В., 2023

Ефремова Светлана Владимировна – ведущий специалист, Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: efremova_svet@sibsau.ru.

Efremova Svetlana Vladimirovna – Leading Specialist, Reshetnev Siberian State University of Science and Technology. E-mail: efremova_svet@sibsau.ru.
