

УДК 629.78

DOI 10.17513/snt.39691

## СПОСОБ ПРОЕКТИРОВАНИЯ ИНТЕГРИРОВАННОЙ БОРТОВОЙ АППАРАТУРЫ КОСМИЧЕСКОГО АППАРАТА С ИСПОЛЬЗОВАНИЕМ АРХИТЕКТУРНОГО ЯЗЫКА AADL

Киреев А.П., Шаров С.А.

*ФГБВОУ ВО «Военно-космическая академия имени А.Ф. Можайского», Санкт-Петербург,  
e-mail: vka@mil.ru*

Авторами предложен алгоритм проектирования бортовой аппаратуры космического аппарата на специализированном предметно-ориентированном языке проектирования встроенных систем AADL (Architecture Analysis and Design Language). Применение традиционных способов разработки бортовой аппаратуры, основанных на строгом описании всех требований и архитектурных решений в текстовых документах при возрастающей технической сложности проектов современных космических аппаратов, становится чрезмерно трудоемким и подверженным ошибкам. Спецификация проекта с использованием универсальной архитектурной модели и ее автоматизированная обработка позволит провести предварительную оценку качества изделия до появления опытного образца. В статье приведен анализ производительности проекта бортовой аппаратуры на ранней стадии процесса проектирования бортовой аппаратуры космического аппарата для определения требуемых вычислительных ресурсов при использовании различных аппаратных конфигураций. По результатам исследований установлено, что увеличение производительности бортовой аппаратуры космического аппарата не приводит к линейному росту производительности специального программного обеспечения обработки радиолокационных данных. Дальнейшее эволюционное развитие проектируемой системы возможно за счет расширения набора используемых формальных моделей различных подсистем, что обеспечит соответствие свойств разрабатываемой системы заданным требованиям и позволит оценить качество принятых технических решений.

**Ключевые слова:** проектирование бортовой аппаратуры, космический аппарат, программное обеспечение, встроенные системы, архитектурная модель, архитектурный язык AADL

## A METHOD FOR DESIGNING AN INTEGRATED ONBOARD EQUIPMENT OF A SPACECRAFT USING AN ARCHITECTURAL LANGUAGE AADL

Kireev A.P., Sharov S.A.

*Military Space Academy named after A.F. Mozhaiskiy, Saint Petersburg, e-mail: vka@mil.ru*

The authors proposed an algorithm for designing spacecraft onboard equipment using a special domain-specific language for designing embedded systems AADL (Architectural Analysis and Design Language). A distinctive feature of the development of onboard equipment, based on strict observance of all safety requirements and architectural solutions in text documents, with the increasing technical complexity of modern spacecraft projects, is the use of labor-intensive and preferable errors. The design specification using the universal architectural model and its automated processing is likely to give a high estimate of the quality of the product before the approval of the prototype. The article presents an analysis of the performance of the onboard equipment design at the initial stage of the onboard equipment design process. According to the results of the research, it was found that an increase in the productivity of onboard space measurement equipment does not lead to a linear increase in the production of low frequency processing of radar data. Further evolutionary development of the designed system is possible by expanding the set of used formal models of various subsystems, which will ensure that the properties of the developed system meet the specified requirements and will allow assessing the quality of the adopted technical solutions.

**Keywords:** onboard equipment design, spacecraft, software, embedded systems, architectural model, AADL architectural language

Основой современных технологий проектирования и разработки сложных технических систем, обеспечивающих высокую надежность и масштабируемость, являются формальные методы, под которыми понимаются математически обоснованные методы спецификации, разработки и верификации аппаратного и программного обеспечения (ПО) технических систем [1, с. 370]. Дальнейшее эволюционное развитие проектируемой системы поддерживается за счет расширения формальных моделей, что обеспечивает соответствие свойств разраба-

тываемой системы заданным требованиям и определяет достоверность принятых технических решений.

При построении современных интегрированных средств проектирования бортовой аппаратуры (БА) космических аппаратов (КА) необходимо соблюдать следующие требования:

– применять методы интегрированной модульной разработки, обеспечивающие проектирование и интеграцию программного и аппаратного обеспечения, использовать архитектурные модели для автоматизации,

а также унификации представления данных разрабатываемого проекта;

- использовать языки описания архитектурных моделей;
- использовать автоматизацию проектирования и представления данных проекта БА, в том числе в среде Заказчика;
- применять методы анализа и синтеза отдельных частей модели, а также верификации сложных технических систем на основе архитектурных моделей.

При решении задач такого типа требуется точная детализация бортовой аппаратуры на различных уровнях представления, а также гарантированная точность и трассируемость требований при анализе последствий в случае внесения изменений в процессе проектирования. При этом возрастающая техническая сложность проектов современных КА требует более тщательного анализа и проектирования. В таких условиях применение специалистами традиционных способов разработки БА, основанных на строгом описании всех требований и архитектурных решений в текстовых документах, становится чрезмерно трудоемким и подверженным ошибкам [2, с. 204].

Возможность подключить к разрешению указанных проблем программные средства автоматизации наталкивается на слабую

структурированность информации, а также ее разнородность в различных проектах.

В связи с этим требуется учитывать спецификацию проекта с использованием универсальной модели для ее автоматизированной обработки.

Из вышеизложенного следует, что необходимо разрабатывать как технологии для оптимизации сложных программно-аппаратных проектов КА, которые позволили бы провести предварительную оценку качества изделия до появления опытного образца, так и его анализ на отказоустойчивость. В авиационной промышленности примерами таких решений являются интегрированные среды OSATE [3, с. 259], MASIW [4].

В данной статье представлено решение задачи применения архитектурного языка AADL (Architecture Analysis and Design Language), который в настоящее время является стандартом для моделирования архитектуры и выполнения требований по безопасности БА в аэрокосмической области [5], при проектировании системы бортовой аппаратуры.

Целью исследования является разработка способа проектирования интегрированной бортовой аппаратуры космического аппарата с использованием архитектурного языка AADL.



Рис. 1. Структурная схема применения формальных методов для спецификации моделей требований и поведения

## Материалы и методы исследования

### *Применение способа проектирования интегрированной бортовой аппаратуры космического аппарата с использованием архитектурного языка AADL*

Основой современного проектирования аппаратного и программного обеспечения бортовой аппаратуры космических аппаратов является применение формальных методов.

При проектировании и разработке систем формальные методы обычно включают следующие три вида деятельности [6]:

1. Системное моделирование – абстрактное представление аппаратного и программного обеспечения для анализа, моделирования и генерации кода.

2. Формальная спецификация – описание некоторых свойств, которым должна удовлетворять система.

3. Формальная верификация – проверка того, что модель системы удовлетворяет спецификации.

Необходимо отметить, что архитектурное проектирование является первоначальным этапом применения формальных методов при создании системы. Как показано на рис. 1, основная идея формального анализа требований к проектированию системы состоит в том, чтобы специфицировать модели поведения и требований бортовой аппаратуры, подлежащие их проверке с использованием выбранного формального научно-методического аппарата.

В настоящее время существуют определенные инструменты разработки технических систем, которые обеспечивают не только среду моделирования, но также и определенные встроенные модели их верификации на базе формальных методов. Наиболее известной средой разработки является программный продукт OSATE, который охватывает этапы разработки от требований к системе до программного кода, включая спецификацию требований, моделирование встроенных систем, оценку отказоустойчивости и т.п. [7].

На основе вышеизложенного в статье представлен способ проектирования интегрированной бортовой аппаратуры космического аппарата с использованием архитектурного языка AADL.

### *Способ проектирования интегрированной бортовой аппаратуры космического аппарата с использованием архитектурного языка AADL*

Предметно-ориентированный архитектурный язык AADL предназначен для построения моделей технических систем, ко-

торые используются для представления статической модульной архитектуры программного обеспечения, архитектуры среды выполнения с точки зрения решения коммуникационных задач, а также архитектуры компьютерной платформы, на которой программное обеспечение развертывается, и любой физической системы или среды, с которыми система взаимодействует.

В языке AADL используются следующие встроенные средства для формализованного описания элементов системы и отношений между ними:

- на аппаратном уровне: элементарное устройство, процессор, память, порт, шина;
- на абстрактном уровне: виртуальный процессор, виртуальная шина, поток исполнения.

При разработке проекта БА его архитектура представляется как иерархия взаимодействующих компонентов. При этом разрабатываются спецификации интерфейсов, схемы реализации программного обеспечения, а аппаратные и физические компоненты группируются в пакеты для поддержки масштабирования и совместной работы проектировщиков.

Свойства компонентов в языке AADL представляют собой универсальный механизм семантического расширения конструкций языка, так как могут вводиться пользователем в соответствии с абстракциями создаваемой модели практически без ограничений.

Алгоритм проектирования интегрированной БА КА с использованием архитектурного языка AADL представлен на рис. 2.

На первом этапе работы осуществляется определение компонентов модели БА и размещение их спецификаций в пакеты.

На следующем этапе функционирования алгоритма производится интеграция отдельных компонентов в систему (разработка модели верхнего уровня) путем объявления экземпляров компонентов и их связей. При этом определяются окончательный вариант системы и ее реализация. Компоненты проектируемой системы могут быть аппаратными устройствами (например, датчики, устройства управления или интерфейсные блоки оператора) или представлять собой программное обеспечение.

Затем производится детализация состава процессов управления, которая включает в себя принятие решений, связанных с разделением функциональности, требуемых от БА для обеспечения контроля управления. Для каждого процесса управления создаются потоки исполнения, которые обеспечивают его функциональность.

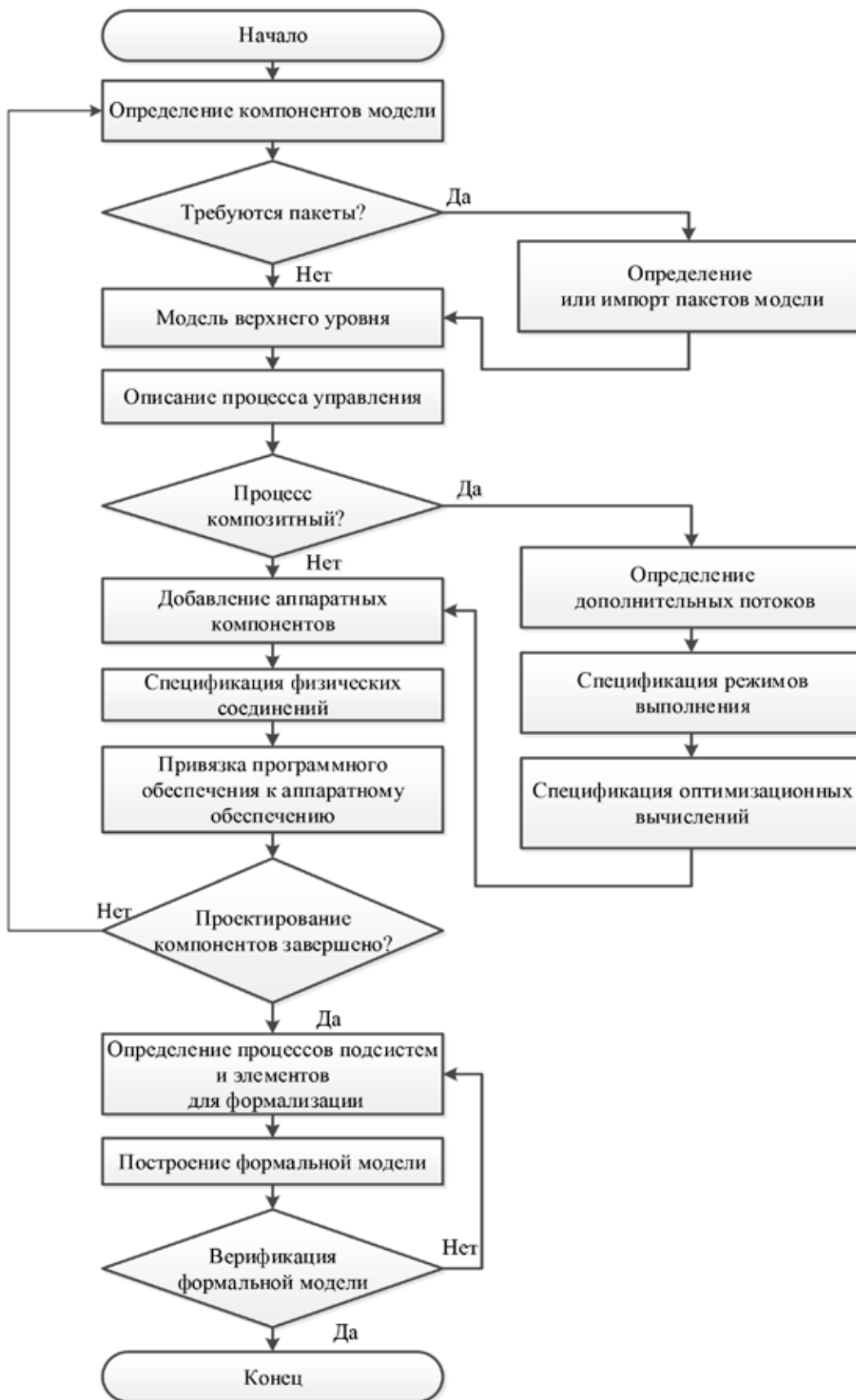


Рис. 2. Схема алгоритма проектирования БА КА на языке AADL

Потоки могут получать данные от датчиков, выполнять операции масштабирования и фильтрации данных, пересылки данных в другие потоки. В случае композитного решения для процесса управления определяются дополнительные потоки, осуществляется спецификация их режимов выполнения и оптимизации вычислений.

Следующим этапом является добавление аппаратных компонентов алгоритма. После определения компонентов программного обеспечения системы управления производится определение аппаратного обеспечения среды выполнения, необходимое для поддержки ПО.

На этапе объявления физических соединений алгоритма продолжается интеграция проекта, добавляются экземпляры, необходимые для компонентов платформы исполнения реализации системы. Так как управляющее программное обеспечение системы, выполняющееся на процессоре, должно получать данные от датчиков и интерфейсных блоков, а также отправлять команды к исполнительным устройствам, то необходимо создать объявление, связывающее исполнительные устройства и шину.

Следующим этапом является привязка программного обеспечения к аппаратному обеспечению. Объявления в проекте позволяют специфицировать потокам исполнения определенный тип процессоров (для решения сложных вычислительных задач, например для сильно загруженного потока исполнения в современном встроенном оборудовании может быть указан тип процессора 500 или 1000 МГц). Также проводится анализ планирования и оценивается загрузка процессора потоками исполнения.

По построенной модели системы и подсистем строится формальное описание для верификации требуемых свойств разрабатываемой системы.

Рассмотрим применение формальных методов для верификации созданных архитектурных моделей. Авторами статьи в докладе [8] рассмотрена верификация протокола информационно-логического взаимодействия бортового комплекса управления (БКУ) и специальной аппаратуры с помощью формальной модели. Для проверки выполнимости свойств протокола информационного взаимодействия задаются формулы линейной темпоральной логики (LTL).

Пример проверки ограничения между двумя условиями  $Q$  и  $R$ :

$$G((Q \vee !R \wedge FR) \rightarrow (!P \cup R)), \quad (1)$$

где  $Q$  – БКУ провел тестирование,  
 $P$  – PCA прошел режим тестирования аппаратуры,

$R$  – БКУ передал команду на установку режима радиолокационного наблюдения.

Предложенный способ ранней диагностики ошибок СПО БА КА, основанный на темпоральной логике и общей формализованной модели данных, предоставляет возможность раннего обнаружения ошибок функционирования информационных протоколов программного обеспечения бортовой аппаратуры КА.

Другой областью применения формальных методов при проектировании бортовой аппаратуры является верификация задач систем реального времени (СРВ) [9, с. 454]. Планировать задачи в СРВ можно статическими и динамическими методами. В качестве политики планировщика со статическими приоритетами используется алгоритм RMS (Rate Monotonic Scheduling – планирование с монотонной скоростью). Статические приоритеты назначаются задачам в соответствии с продолжительностью цикла задания, поэтому наиболее короткая продолжительность цикла приводит к более высокому приоритету задачи. В случае применения планировщиком алгоритма EDF (Earliest Deadline First – первоочередное планирование более раннего срока выполнения) используется динамический принцип назначения приоритетов задачам. Планировщик задач, построенный на данном алгоритме, считает, что чем ближе предельный срок выполнения задачи, тем ее приоритет в системе выше.

Для описания свойств планировщика задач в СРВ был использован механизм дополнений языка AADL – Cheddar [10, с. 4], позволяющий расширять функциональную содержательность проектируемой архитектуры.

При моделировании приложения реального времени представляются набором задач. Для планирования времени исполнения каждая задача обычно определяется следующими временными параметрами [11, с. 7]:

- $C_i$  – время выполнения задачи в наихудшем случае (worst-case execution time – WCET),
- $P_i$  – период (продолжительность между двумя задачами),
- $D_i$  – критический срок завершения выполнения задачи;
- $U$  – коэффициент использования процессора.

Необходимое условие того, что система реального времени будет исполнимой – коэффициент использования процессора должен быть меньше или равен 1:

1) по алгоритму RMS

$$U = \sum_{i=1}^n \frac{C_i}{P_i}; \quad (2)$$

2) по алгоритму EDF

$$U = \sum_{i=1}^n \frac{C_i}{D_i}. \quad (3)$$

В качестве примера планирования проекта рассмотрим варианты оборудования КА в составе: три потока исполнения (поток управления `thread_control`, поток обработки данных `thread_data` и поток управления радиоканалом `thread_radio`) и один или два процессора (с тактовой частотой 1 ГГц).

При этом потоки исполнения задаются с конкретным периодом выполнения: поток `thread_data` – 2, ..., 18 мс, поток `thread_radio` – 1, ..., 6 мс, поток `thread_control` – 13, ..., 34 мс.

В табл. 1 представлены оценки производительности (общей загрузки вычислительных ресурсов) бортовой аппаратуры с двумя вариантами.

### Результаты исследования и их обсуждение

Из анализа полученных результатов следует, что в случае применения одного процессора общая загрузка потоков превышает вычислительную производительность системы. Поэтому нет возможности построить корректное расписание, что приводит к срыву выполнения задачи обработки (потоки управления имеют больший приоритет в бортовых вычислительных системах). При варианте с двумя процессорами потоки

используют 62 % вычислительных ресурсов проекта БА, что позволяет организовать дополнительные задачи обработки или передачи данных.

Оценки планирования потоков исполнения могут основываться на временных измерениях кода прототипа или данных для аналогичного проекта (например, системы с одинаковыми или сопоставимыми процессорами). Проводя анализ на ранней стадии процесса разработки с использованием технологии AADL, можно получить количественные прогнозы производительности проекта.

С целью определения требуемых вычислительных ресурсов для проектируемого КА радиолокационного наблюдения были проведены расчеты для различных реализаций алгоритмов по типовым операциям обработки радиолокационных данных – свертки радиолокационного изображения (РЛИ) по дальности и азимуту (табл. 2).

Анализ оценок производительности алгоритмов синтеза РЛИ показывает наибольший прирост при многопоточной реализации для свертки по дальности.

Как видно из представленных данных, увеличение производительности бортовой аппаратуры космического аппарата не приводит к линейному росту производительности специального программного обеспечения обработки радиолокационных данных.

Таблица 1

Предварительная оценка производительности бортовой аппаратуры

Потоки системы	Вариант системы (1 процессор)	Вариант системы (2 процессора)
Поток обработки данных ( <code>thread_data</code> )	2–18 мс	2–18 мс
Поток управления ( <code>thread_control</code> )	13–34 мс	13–34 мс
Поток радиоканала ( <code>thread_data</code> )	1–6 мс	1–6 мс
Общая загрузка	Превышение вычислительной производительности системы	62,46 %

Таблиц 2

Оценка производительности алгоритмов обработки радиолокационных данных

Наименование операции	Размеры исходного файла, пиксели	Время реализации варианта обработки, с	
		Обычный (1 процессор)	Многопоточный (2 процессора)
Свертка РЛИ (дальность)	20 × 23552	15,72	11,04
Свертка РЛИ (азимут)	20 × 23552	5,28	4,48
Свертка РЛИ (дальность)	200 × 23552	140,05	98,03
Свертка РЛИ (азимут)	200 × 23552	76,35	64,89

Кроме указанных операций синтеза, на производительность обработки радиолокационных данных большое влияние оказывает выбор вычислительной платформы и способа реализации алгоритмов. Располагая на борту стандартизированной вычислительной платформой задачи оптимизации кода являются для данных средств приоритетными (в том числе низкоуровневое исполнение на языках Си и ассемблере и использование специализированных плат обработки).

Таким образом, приведенные оценки производительности проекта помогают избежать проблем на поздних этапах его создания (например, во время реализации кода или системной интеграции). Разработка БА КА с использованием архитектурного языка проектирования AADL позволяет автоматизировать процесс разработки бортовой аппаратуры, производить предварительное планирование потоков исполнения, использовать библиотеки компонентов проектов для стандартизованных решений, представлять в среде Заказчика данные по текущему состоянию проекта БА КА с целью оценки реального прогресса по работам и оценки обоснованности технических решений.

### Заключение

Применение универсальных архитектурных языков при проектировании бортовых систем КА является необходимым условием создания современных интегрированных решений в космической отрасли.

В данной статье рассмотрен способ проектирования интегрированной БА КА с использованием архитектурного языка AADL. Представленный алгоритм основан на применении архитектурных методов разработки бортового комплекса КА, определяет процесс автоматизированного проектирования и интеграции программного и аппаратного обеспечения.

Результаты проведенных исследований могут быть использованы при создании но-

вых образцов космической техники и разработки процедур верификации проектов интегрированной бортовой аппаратуры КА.

### Список литературы

1. Lee E.A., Seshia S.A. Seshia. Introduction to Embedded Systems – A Cyber-Physical Systems Approach. Second Edition. MIT Press. 2017. P. 565.
2. Буздалов Д.В., Зеленов С.В., Корныхин Е.В., Петренко А.К., Страх А.В. Инструментальные средства проектирования систем интегрированной модульной авионики // Труды Института системного программирования РАН. 2014. Т. 26. С. 201–230.
3. Зеленова С.А., Зеленов С.В. Моделирование программно-аппаратных систем и анализ их безопасности // Труды Института системного программирования РАН. 2017. Т. 29. Вып. 5. С. 257–282. DOI: 10.15514/ISPRAS-2017-29(5)-13.
4. MASIW: Modular Avionics System Integrator Workplace. [Электронный ресурс]. URL: <https://forge.ispras.ru/projects/masiw/> (дата обращения: 20.02.2023).
5. A Requirement Specification Language for AADL [Электронный ресурс]. URL: [https://www.researchgate.net/publication/305849683\\_A\\_Requirement\\_Specification\\_Language\\_for\\_AADL](https://www.researchgate.net/publication/305849683_A_Requirement_Specification_Language_for_AADL) / (дата обращения: 14.03.2023).
6. Zongyu Cao, Wanyou Lv, Yanhong Huang, Jianqi Shi, Qin Li. Formal Analysis and Verification of Airborne Software Based on DO-333. Electronics. 2020. № 9 (2). 327. DOI: 10.3390/electronics9020327.
7. OSATE: Open Source AADL2 Tool Environment [Электронный ресурс]. URL: <http://osate.org/> (дата обращения: 03.04.2023).
8. Киреев А.П., Санин М.Д., Шаров С.А. Способ раннего диагностирования функциональных ошибок программного обеспечения бортовой аппаратуры космических аппаратов // Актуальные проблемы создания космических систем дистанционного зондирования Земли: материалы девятой международной научно-технической конференции. М.: Труды ВНИЭМ. Вопросы электромеханики, 2021. С. 102–108.
9. Giorgio C. Buttazo. Hard real-time computing systems: predictable scheduling algorithms and applications. Kluwer academics. Springer Science Business Media. LLC 2011. P. 4. С. 521.
10. Frank Singhoff, Alain Plantec, Stéphane Rubini, Vincent Gaudel, Shuai Li1, Christian Fotsing, Laurent Lemarchand, Pierre Dissaux, Jérôme Legrand. How architecture description languages help schedulability analysis: a return of experience from the Cheddar project. Brest, France. 2019. P. 39.
11. Christian Fotsing, Frank Singho, Alain Plantec, Vincent Gaudel, Stéphane Rubini, Shuai Li, Hai Nam Tran, Laurent Lemarchand. Cheddar Architecture Description Language, Brest, France, November 20, 2014. P. 64.