УДК 004.05

А.В. Самонов, С.Р. Малышев, С.В. Краснов, А.Л. Савкин

МЕТОДЫ И СРЕДСТВА РЕАЛИЗАЦИИ ПРОЦЕССА СКВОЗНОГО КОНТРОЛЯ КАЧЕСТВА АРТЕФАКТОВ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ КРИТИЧЕСКИ ВАЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Самонов Александр Валерьянович, *кандидат технических наук, доцент, окончил Военную инженерную Краснознаменную академию им. А.Ф. Можайского. Старший научный сотрудник Военно-космической академии им. А.Ф. Можайского. Имеет более 30 трудов в области системного анализа, системной и программной инженерии, валидации и верификации программного обеспечения, методов и средств обеспечения информационной безопасности. [e-mail: a.samonov@mail.ru].*

Малышев Сергей Романович, *кандидат технических наук, доцент, окончил Военный инженерный Краснознаменный институт им. А.Ф. Можайского. Заслуженный изобретатель РФ. Доцент Военной академии связи (ВАС) им. С.М. Буденного. Имеет учебные пособия, статьи и изобретения в области современных теоретических аспектов ведения радиомониторинга и радиотехнического контроля. [e-mail: malishevsr56@ya.ru].*

Краснов Сергей Васильевич, *кандидат технических наук, доцент, окончил Ульяновское высшее военное командное училище связи (УВВКУС), адъюнктуру Ульяновского высшего военного инженерного училища связи. Доцент Высшей школы управления и бизнеса Санкт-Петербургского политехнического университета Петра Великого. Имеет более 50 работ в области проектирования и внедрения информационных систем и технологий. [e-mail: hsm.krasnov@gmail.com].*

Савкин Александр Леонидович, *кандидат военных наук, доцент, окончил УВВКУС, ВАС им. С.М. Буденного, адъюнктуру там же. Начальник службы обеспечения научно-технической деятельности ФНПЦ АО «НПО «Марс». Имеет научные работы, учебные пособия, статьи в области разработки и моделирования систем управления и связи, автоматизированных информационных систем. [e-mail: mars@mv.ru].*

Аннотация

Представлены методы и средства реализации последовательно-итерационного, программно-управляемого процесса разработки и верификации комплекса требований и проектных решений, разрабатываемых при создании автоматизированных информационных систем, используемых на объектах критической инфраструктуры государства. Дано описание единой модельно-языковой и информационно-программной среды, разработанной на основе языков моделирования и проектирования SysML, FUML, OCL, математического аппарата сетей Петри, временных автоматов, а также программных средств верификации и анализа формальных моделей комплекса требований и проектных решений CPN Tools, Rodin и SPIN. Построение формальных графово-текстовых моделей комплекса требований и проектных решений выполняется на основе разработанных для этого шаблонов, представляющих собой расширения диаграмм вариантов использования, деятельности, состояний и взаимодействия, внешнего и внутреннего представлений компонентов системы, активных и пассивных классов.

Процедура тестирования полноты и корректности комплекса требований осуществляется посредством анализа хода и результатов исполнения его формальной модели в среде виртуальной машины VM FUML, верификации поведенческой части модели с помощью CPN Tools и Rodin на предмет стабильности, ограниченности, выявления тупиковых и недоступных состояний, а также посредством оценивания качества реализации базовых механизмов объектно-ориентированной парадигмы (инкапсуляции, наследования, полиморфизма, абстракции, обмена сообщениями). Формальные модели проектных решений, представленные в виде диаграмм активностей, транслируются на язык CPN ML и затем верифицируются средствами CPN Tools. Диаграммы последовательностей транслируются на язык Promela и верифицируются в среде SPIN. Диаграммы состояний транслируются на язык EventB и проверяются с помощью средств Rodin.

Реализация представленного в статье подхода создания критически важных информационных систем позволит осуществлять раннее обнаружение и устранение дефектов требований и проектных решений непосредственно на этапах их разработки, что приведет к повышению качества разрабатываемых систем и снижению финансовых и временных затрат, связанных с выполнением дополнительных работ.

Ключевые слова: валидация и верификация, временные автоматы, информационные системы, проектирование и моделирование, сети Петри, функциональные и эксплуатационные требования.

doi: 10.35752/1991-2927-2019-3-57-12-20

METHODS AND MEANS FOR IMPLEMENTATION OF END-TO-END QUALITY CONTROL OF ARTIFACTS OF THE LIFE CYCLE OF CREATING CRITICAL INFORMATION SYSTEMS

Aleksandr Valerianovich Samonov, *Candidate of Science in Engineering, Associate Professor; graduated from Mozhaisky Red Banner Military Engineering Academy; Senior Scientist of Mozhaisky Military Space Academy; an author of more than 30 publications in the field of system analysis, system and software engineering, validation and verification of software, methods and means of information security. e-mail: a.samonov@mail.ru.*

Sergei Romanovich Malyshev, *Candidate of Science in Engineering, Associate Professor; graduated from the Mozhaisky Military Institute of Engineering; Honored Inventor of the Russian Federation; Associate Professor of the Marshal Budjonny Military Academy of Signal Corps; an author of manuals, articles, and inventions in the field of modern theoretical aspects of radiomonitoring. e-mail: malishevsr56@ya.ru.*

Sergei Vasilevich Krasnov, *Candidate of Science in Engineering, Associate Professor; graduated from the Ulyanovsk Higher Military Command School of Communications; an adjunct of Ulyanovsk Higher Military Engineering Communication School; Associate Professor of Graduated School of Management and Business of Peter the Great St.Petersburg Polytechnic University (SPbPU); an author of more than 50 works in the field of design and implementation of information systems and technologies. e-mail: hsm.krasnov@gmail.com.*

Aleksandr Leonidovich Savkin, *Candidate of Military Sciences, Associate Professor; graduated from the Ulyanovsk Higher Military Command School of Communications and the Marshal Budjonny Military Academy of Signal Corps; completed postgraduate studies in the Military Academy of Communications; Head of the Science and Engineering Support Department of FRPC JSC 'RPA 'Mars'; an author of scientific works, manuals, and articles in the field of the development and modeling of communication control systems, automated information systems. e-mail: mars@mv.ru.*

Abstract

Methods and means of realization of the sequential-iterative, program-controlled process of development and verification of a set of requirements and design decisions developed at creation of the automated information systems used on objects of critical infrastructure of the state are presented. The description of the unified model-language and information-software environment developed on the basis of modeling and design languages SysML, FUML, OCL, mathematical modeling language Petri nets, timed automata as well as software tools for verification and analysis of formal models of complex requirements and design solutions CPN Tools, Rodin and SPIN. The construction of formal graph-text models of requirements and design solutions is based on the developed templates, which are extensions of diagrams of use cases, activities, states and interactions, external and internal representation of system components, active and passive classes.

Testing procedure of completeness and correctness of a set of requirements is carried out by analyzing the progress and results of execution its formal model in the VM FUML environment, verification of the behavioral part of the model with the help of CPN Tools and Rodin for stability, limitation, detection of dead-end and inaccessible states as well as by assessing the quality of implementation of the basic mechanisms of the object-oriented paradigm (encapsulation, inheritance, polymorphism, abstraction, messaging). Formal models of design solutions, presented in the form of activity diagrams, are translated into the language of CPN ML and then verified by means of CPN Tools. Sequence diagrams are translated into Promela language and verified in SPIN environment. State diagrams are translated into the EventB language and checked using Rodin tools.

The implementation of the approach presented in the article will allow to carry out early detection and elimination of defects in requirements and design solutions directly at the stages of their development, which will improve the quality of the developed systems and reduce the financial and time costs associated with the implementation of additional work.

Key words: validation and verification, timed automata, information systems, design and modeling, Petri nets, functional and operational requirements.

Введение

Основными компонентами критической информационной инфраструктуры (КИИ) государства являются информационно-телекоммуникационные сети и автоматизированные информационные системы, предназначенные для решения задач государственного управления, обеспечения обороноспособности, безопасности и правопорядка [1]. Нарушение функционирования этих систем, которые в дальнейшем будем называть критически важными информационными системами (КВИС), может привести к самым губительным последствиям для безопасности государства, жизни и здоровья людей. Корректное и надежное функционирование КВИС должно обеспечиваться на всех этапах их жизненного цикла (ЖЦ): от обоснования требований и разработки до эксплуатации и сопровождения. Особая роль и исключительно важное значение при этом имеет качество комплекса требований и проектных решений, разрабатываемых на первых двух этапах ЖЦ систем. При разработке КВИС необходимо учитывать следующие особенности как самих систем, так и процесса их создания:

- отсутствие прямых аналогов, ограничивающее возможность использования типовых проектных решений и готовых прикладных систем;
- высокие требования к их надежности, производительности и защищенности;
- необходимость интеграции вновь разрабатываемых систем с существующими и унаследованными;
- функционирование в неоднородной среде на нескольких аппаратных платформах.

Использование некачественных требований и проектных решений приводит не только к увеличению временных и финансовых затрат, но и к получению системы, не соответствующей потребностям пользователя. Данный факт наглядно подтверждают результаты анализа проектов по созданию информационных систем, ежегодно публикуемые американской компанией The Standish Group. В отчете за прошлый год утверждается, что доля успешных проектов составляет около 30 процентов, при этом 20 процентов проектов провалились, остальные 50 процентов проектов столкнулись с проблемами, из-за которых итоговый бюджет превысил первоначальный в среднем в 1,5 раза, сроки выросли почти в 2 раза, около 50 процентов функций не соответствуют заявленным требованиям [2].

Одной из основных причин такого состояния дел является наличие логических и семантических разрывов между представлениями о разрабатываемой системе главных участников данного процесса: заказчика и пользователей, конструкторов и проектировщиков, инженеров и программистов. Это приводит к тому, что связи между тремя основными артефактами ЖЦ разработки КВИС являются несистемными, фрагментарными, неточными и несогласованными.

В связи с вышеизложенным предлагается для снижения влияния этих негативных факторов и устранения порождаемых ими проблем разработать и предоставить всем участникам процесса создания КВИС единую модельно-языковую и информационно-программную среду, в которой реализовать последовательно-итерационный, программно-управляемый процесс разработки и контроля качества всех артефактов ЖЦ создания системы: комплекса требований, проектных решений и реализации. В следующих разделах статьи представлены использованные и разработанные для решения данной задачи математические модели, языки и средства моделирования и верификации.

Математические модели и средства формального представления комплекса требований и проектных решений КВИС

При разработке единой модельно-языковой и информационно-программной среды, в которой можно эффективно реализовать процесс сквозного контроля качества артефактов ЖЦ КВИС, были использованы следующие технологии и методы модельно-ориентированной системной инженерии (МОСИ) (Model-based systems engineering, MBSE): разработка на основе моделей (Model driven development, MDD), архитектура на основе моделей (Model driven architecture, MDA), тестирование на основе моделей (Model based testing, MBT) [3, 4]. Реализация данных технологий и методов предполагает, что создаваемые и используемые на всех этапах ЖЦ системы артефакты (комплекс требований, технический проект, программные и технические реализации) представляются в виде формальных моделей, которые подвергаются процедурам верификации и валидации с помощью специальных автоматизированных методов и средств. По результатам проверки качества этих артефактов принимается решение об их пригодности для использования на следующих этапах ЖЦ системы или направлении на доработку.

Для разработки формальных моделей требований и проектных решений были использованы языки визуального моделирования и проектирования SysML [5, 6], FUML [7], OCL [8], ALF [9]. Формальная верификация этих моделей осуществлялась с помощью следующих средств: CPN Tools [10], Rodin [11], SPIN [12]. Интеграция представленных выше технологий, моделей и средств в рамках единой модельно-языковой и информационно-программной среды позволяет реализовать последовательно-итерационный, программно-управляемый процесс создания КВИС, предоставляющий возможность осуществлять раннее обнаружение и устранение дефектов требований и проектных решений непосредственно на этапах их разработки.

В настоящее время одним из наиболее перспективных и активно используемых языков проектирования сложных автоматизированных информационных систем является SysML. С помощью данного языка можно решать задачи анализа, специфицирования, проектирования и тестирования систем, состоящих из аппаратных средств, программного обеспечения, данных, персонала, процедур, средств и других искусственных и природных

систем. Поддержка SysML в настоящее время реализована в целом ряде систем автоматизированного проектирования, в том числе в IBM Rhapsody, Sparx Enterprise Arhitect, Modelio, Eclipse Papyrus. С помощью SysML можно описать следующие свойства и аспекты КВИС:

- состав и структуру системы посредством диаграмм пакетов (Package), внешнего и внутреннего представления блоков (Block Definition, Internal Block), активных и пассивных классов (Active Class, Passive Class);

- поведение и способы взаимодействия ее компонентов с помощью диаграмм вариантов использования (UseCase), деятельности (Activity), состояний (State Machine) и взаимодействия (Sequience);

- ограничения на эксплуатационные характеристики (производительность, надежность, защищенность и др.) и физические свойства средствами языка OCL;

- связи между структурными, поведенческими, ограничительными и другими аспектами и характеристиками с помощью отношений обобщения (generalization), агрегации (aggregation), композиции (composition), ассоциации (association), зависимости (dependency) и др.

Все КВИС, с точки зрения особенностей алгоритма их функционирования, можно разделить на три группы. Первую группу образуют системы, выполняющие свои функции в соответствии с определенным алгоритмом, имеющим только вход и выход. Во вторую группу входят системы, функционирующие по принципу автоматов, изменяющих свое состояние и выполняющих определенные операции в зависимости от внешних воздействий. Третью группу образуют системы, взаимодействующие посредством обмена сообщений.

Для моделирования и проектирования систем первой группы целесообразно использовать диаграммы деятельности (activity), основанные на математическом аппарате сетей Петри (СП)[13]. СП представляется четверкой $N=(P, T, F, m_l)$ где $P=\{p_1, p_2, ..., p_n\}$ – множество мест; $T=\{t_1, t_2, ..., t_n\}$ – множество переходов, таких что $P \cap T=\varnothing$; $F \subseteq P \times T \times T \times P$ – матрица инцидентности, а $m_l: P \longrightarrow N$ – начальная маркировка. Основными свойствами СП, которые следует использовать для верификации формальных моделей требований и архитектуры, являются: живость (отсутствие блокировок и зацикливаний) и ограниченность. Проверка этих свойств осуществляется посредством анализа достижимости и покрываемости.

Решение проблемы достижимости и покрываемости осуществляется посредством анализа пространства состояний системы и проверки принадлежности выбранных маркировок ко всему пространству состояний. Достижимость – это наличие путей перехода в заданные состояния из начального состояния $m \in [m_l\rangle$. Покрываемость – получение заданных состояний из других состояний $m'' \in [m'\rangle$. СП является k-ограниченной тогда и только тогда, когда все маркировки $m \in [m_l\rangle$ и для всех $p \in P: m(p) \le k$. Частным случаем свойства ограниченности является безопасность. СП (P, T, F, m_l) безопасна, если все маркировки $m \in [m_l\rangle$ и для всех $p \in P: m(p) \le 1$. При разработке моделей требований и архитектуры КВИС, имеющих сложную организационно-техническую структуру, целесообразно использовать раскрашенные временные СП (РВСП) следующего вида: $N=(P, T, C, F, m_l, S, Z,)$ где $C=\{c_1, c_2, \ldots, c_d\}$ – цвета или типы маркеров, $S=(s_1, s_2, \ldots, s_n)$ – вектор временных задержек маркеров в позициях, $Z=(z_1, z_2, \ldots, z_r)$ – вектор времени срабатывания разрешенных переходов.

Важным достоинством использования СП является наличие целого ряда автоматизированных средств верификации построенных с их помощью моделей. Одним из наиболее развитых и эффективных является CPN Tools, который и предлагается использовать в нашем случае для анализа формальных моделей требований и архитектуры КВИС.

Для разработки формальных моделей сложных систем, основной алгоритм функционирования которых заключается в переходах между определенными состояниями, обусловленными внешними воздействиями (стимулами, командами), целесообразно использовать временные автоматные модели или временные автоматы (ВА) [14, 15], описываемые следующим выражением:

$$TA \equiv <S, C, D, R, f, s_0>,$$

где S – конечное множество позиций автомата;

C – конечное множество локальных часов, значения которых возрастают синхронно с реальным временем и могут принимать значения из множества действительных чисел $\Re$;

$f: S \longrightarrow P$ – функция, которая ставит в соответствие каждой позиции $s \in S$ некоторый предикат $p \in P$; в общем случае p – есть логическое выражение из ограничений вида $c \le k$, $c < k$, $c_i - c_j \le k$, $c_i - c_j < k$, где c_i, $c_j \in C$, $k \in \Re$. Переход из позиции $s \in S$ должен быть выполнен до того, как предикат $p = f(s)$ станет ложным, в противном случае автомат прекращает работу;

D – конечное множество действий;

$R \subseteq S \times P \times D \times C \times S$ – множество переходов автомата;

$s_0 \in S$ – начальная позиция автомата.

Для моделирования КВИС, которые состоят из нескольких взаимодействующих компонент, целесообразно использовать сети временных автоматов (СВА).

Таким образом, для разработки моделей и алгоритмов формального описания основных артефактов процесса создания КВИС предлагается использовать следующие математические модели и языки моделирования:

- СП, ВА;

- языки визуального моделирования и проектирования SysML, FUML, OCL, ALF.

В качестве автоматизированных средств верификации формальных моделей комплекса требований и проектных решений были использованы: VM FUML,

CPN Tools, SPIN, Rodin. На основе представленных выше моделей, языков и средств создается единая для всех участников разработки КВИС модельно-языковая и информационно-программная среда и реализуется последовательно-итерационный, программно-управляемый процесс проектирования и разработки.

Методика и средства разработки и верификации формальной модели комплекса требований

Комплекс требований к КВИС можно разделить на две основные группы: требования к ее функциональным возможностям и требования к качеству и условиям их реализации. Для обеспечения автоматизированного построения формальных описаний этих требований в среде визуального моделирования были разработаны специальные шаблоны (стереотипы), представляющие собой расширения диаграмм вариантов использования (UseCase), деятельности (Activity), состояний (State Machine) и взаимодействия (Sequence), внешнего (block definition diagram) и внутреннего (internal block diagram) представлений компонентов системы, активных (Active Class) и пассивных (Passive Class) классов.

В частности, для формального описания требований к функциональным возможностям самого верхнего уровня их представления разработан шаблон «function requirement», который является расширением SysML-диаграммы вариантов использования – Use Case diagram (табл. 1).

Каждый i-й вариант использования представляет собой функциональное требование и описывается следующим образом:

$$uc_i = (nameuc_i,\ description_i,\ actor_k,\ subject_j,\ basic\_scenario_i,\ alter\_scenario_i).$$

Функциональные блоки системы (subject) описываются посредством диаграмм пакетов (Package), внешнего и внутреннего представлений блоков (Block Definition, Internal Block). Основной и альтернативные сценарии реализации функций представляются в форме активных и пассивных классов (Active Class, Passive Class). Для описания методов классов используются диаграммы деятельности (Activity), состояний (State Machine) и взаимодействия (Sequience).

Следующим шагом построения модели анализа является разработка диаграммы нефункциональных требований, определенных в техническом задании для каждой функции:

$$d\_reqs = \left(r_1^{f_i},\ r_2^{f_i}, r_3^{f_i}, r_4^{f_i} \ldots\right),$$

где $r_1^{f_i}$ – требования к оперативности исполнения функции f_i;

$r_2^{f_i}$ – требования к производительности (например, объемы хранимых, обрабатываемых и передаваемых данных, количество пользователей, количество и размеры запросов в единицу времени и др.);

$r_3^{f_i}$ – требования к надежности (коэффициент готовности, время безотказной работы, время восстановления и др.);

$r_4^{f_i}$ – требования к защищенности.

Ограничения на физические свойства и эксплуатационные характеристики (производительность, надежность, защищенность и др.) описываются на языке OCL. Для каждой из этих характеристик также разработаны соответствующие шаблоны, которые используются в подключаемых к инструменту Eclipse Papyrus модулях (plugins) для реализации программно-управляемого процесса их формального описания и верификации. Связи между структурными, поведенческими, ограничительными и другими аспектами и характеристиками описываются с помощью отношений обобщения (generalization), агрегации (aggregation), композиции (composition), ассоциации (association), зависимости (dependency) и др.

Формальная модель комплекса требований подвергается процедурам валидации и верификации. Процедура валидации заключается в оценивании комплекса требований на предмет его полноты и корректности и выполняется как программными средствами, так и неформальной экспертизой специалистов в данной предметной области. При верификации проверяются такие свойства модели как непротиворечивость, системность, неизбыточность, безопасность, живость, отсутствие взаимоблокировок, невыполнимых операций, зацикливаний. Для автоматизированного выполнения процедур

Таблица 1

Шаблон представления требований к функциональным возможностям КВИС

Поле шаблона	Тип данных	Описание поля шаблона	Обязательность задания
nameuc	string	имя (идентификатор требования)	обязательное
description	string	краткое описание функции	обязательное
actor	actor (actors)	пользователь функции	обязательное
subject	class (block)	класс (блок, модуль) системы, реализующий функцию	обязательное
basic_scenarios	scenarios (control&object_flow)	основной сценарий выполнения функции	опциональное
alter_scenarios	scenarios (control&object_flow)	альтернативный сценарий выполнения функции	опциональное

верификации и валидации формальной модели комплекса требований к КВИС разработано соответствующее модельное, информационное, алгоритмическое и программное обеспечение. Представленная на рисунке 1 схема иллюстрирует порядок их применения для выполнения следующих основных методов тестирования комплекса требований:

- валидация комплекса требований посредством исполнения его формальной модели в среде виртуальной машины VM FUML и анализа ее состояния в контрольных точках и при возникновении критичных событий;
- анализ поведенческой части модели, представленной диаграммами активности, с помощью CPN Tools на предмет выявления тупиковых и мертвых состояний, стабильности, ограниченности и безопасности;
- анализ поведенческой части модели, представленной диаграммами состояний, с помощью инструмента Rodin на предмет выявления тупиковых и мертвых состояний, необрабатываемых стимулов и др.;
- верификация качества реализации в диаграмме классов базовых механизмов объектно-ориентированной парадигмы (инкапсуляции, наследования, полиморфизма, абстракции, обмена сообщениями) посредством расчета и оценивания метрик, характеризующих эти механизмы.

В случае обнаружения каких-либо дефектов в комплекс требований вносятся необходимые исправления, и он подвергается повторной процедуре валидации и верификации. После прохождения описанной в данном разделе процедуры тестирования комплекс функциональных и эксплуатационных требований будет обладать следующими необходимыми для его использования на этапе проектирования КВИС свойствами: полнотой, корректностью, непротиворечивостью, системностью, однозначностью, согласованностью. Такие свойства комплекса требований как прослеживаемость, верифицируемость и модифицируемость обеспечиваются средствами единой модельно-языковой и информационно-программной среды, в рамках которой осуществляется разработка всех артефактов ЖЦ КВИС.

Методика и средства разработки и верификации архитектуры и проектных решений КВИС

Формальная модель проектных решений КВИС создается на основе разработанного ранее комплекса требований в той же среде визуального моделирования и проектирования. Для реализации программно-управляемого процесса ее построения было разработано соответствующее модельное, информационное, алгоритмическое и программное обеспечение. Построение формальной модели проектных решений осуществляется путем заполнения предоставляемых в определенном порядке специальных шаблонов SysML-диаграмм требуемой (установленной) информацией. Шаблоны разработаны на основе следующих SysML-диаграмм: внешнего (bdd, block definition diagram) и внутреннего

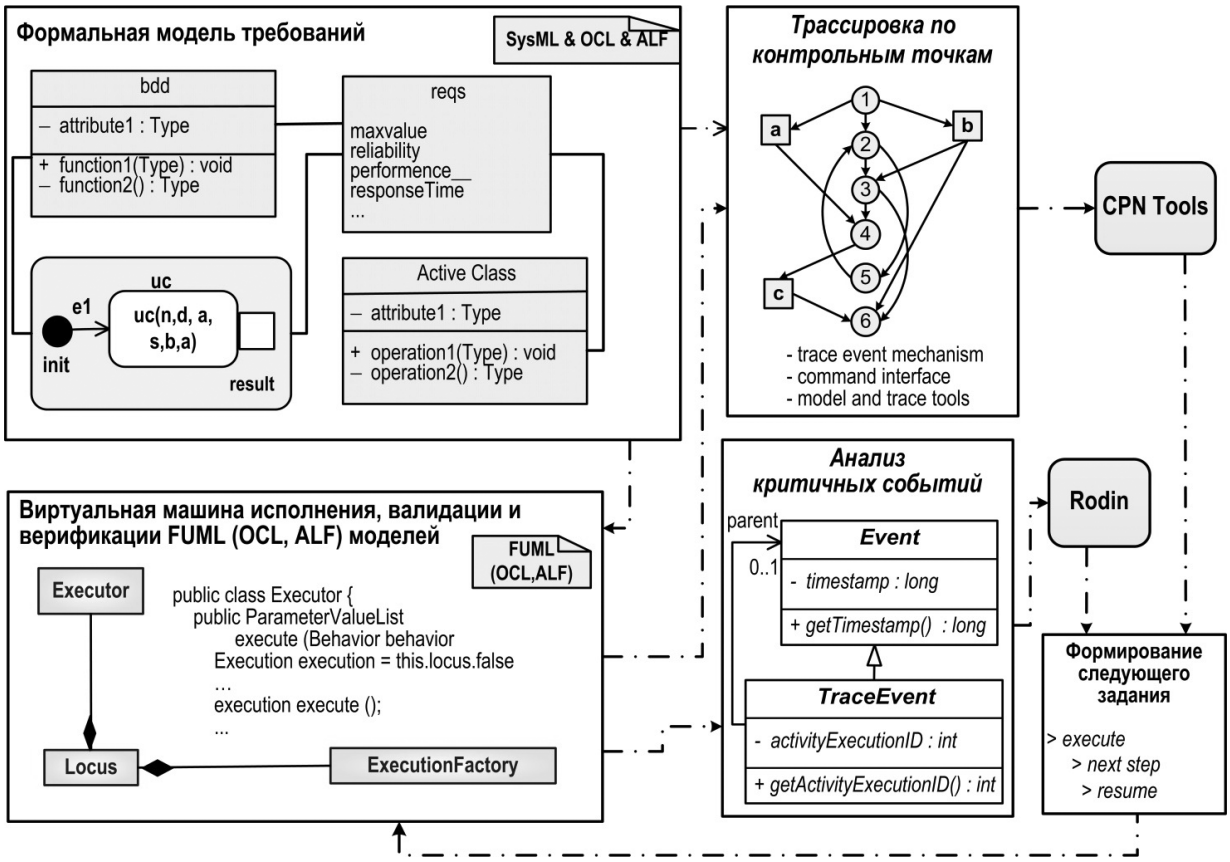

Рис. 1. Модельное, информационное, алгоритмическое и программное обеспечение реализации процедур верификации и валидации формальной модели требований к КВИС

(ibd, internal block diagram) представления компонентов системы, активных (active_class) и пассивных классов (passive_class), диаграмм деятельности (Activity), машин состояний (State Machine) и взаимодействия (Sequence). Для представления отношений между ними и описания эксплуатационно-технических характеристик проектируемой системы используются язык OCL и библиотека Eclipse OCL Standard Library [16]. Формальная модель архитектуры и проектных решений должна удовлетворять следующим характеристикам качества:

- полнота и корректность реализации функциональных требований;
- полнота и корректность реализации нефункциональных требований;
- согласованность и непротиворечивость всех диаграмм модели;
- отсутствие избыточности диаграмм и их атрибутов;
- отсутствие взаимоблокировок, невыполнимых операций и зацикливаний, которые могут привести к нарушению безопасности и живости системы.

Процедура проверки непротиворечивости, согласованности и неизбыточности, входящих в проект КВИС диаграмм, заключается в проверке следующих условий:

- каждый класс представлен не менее чем в одной поведенческой диаграмме;
- для каждого метода класса в проекте имеется одна единственная диаграмма поведения (активности, состояний, последовательности);
- сигнатуры методов класса и описывающих их реализацию диаграмм поведения совпадают;
- каждая поведенческая диаграмма имеет связь (отношение realization) с методом класса, который она реализует;
- все действующие лица (actor) и информационные объекты из поведенческих диаграмм должны быть представлены в виде классов (class);
- типы данных элементов диаграмм, связанных отношениями агрегации и композиции, должны совпадать.

Состав и структура комплекса программных средств, разработанных для выполнения автоматизированной проверки этих характеристик, представлена на рисунке 2. Основным методом проверки полноты и корректности реализации функциональных требований в проектных решениях является выполнение формаль-

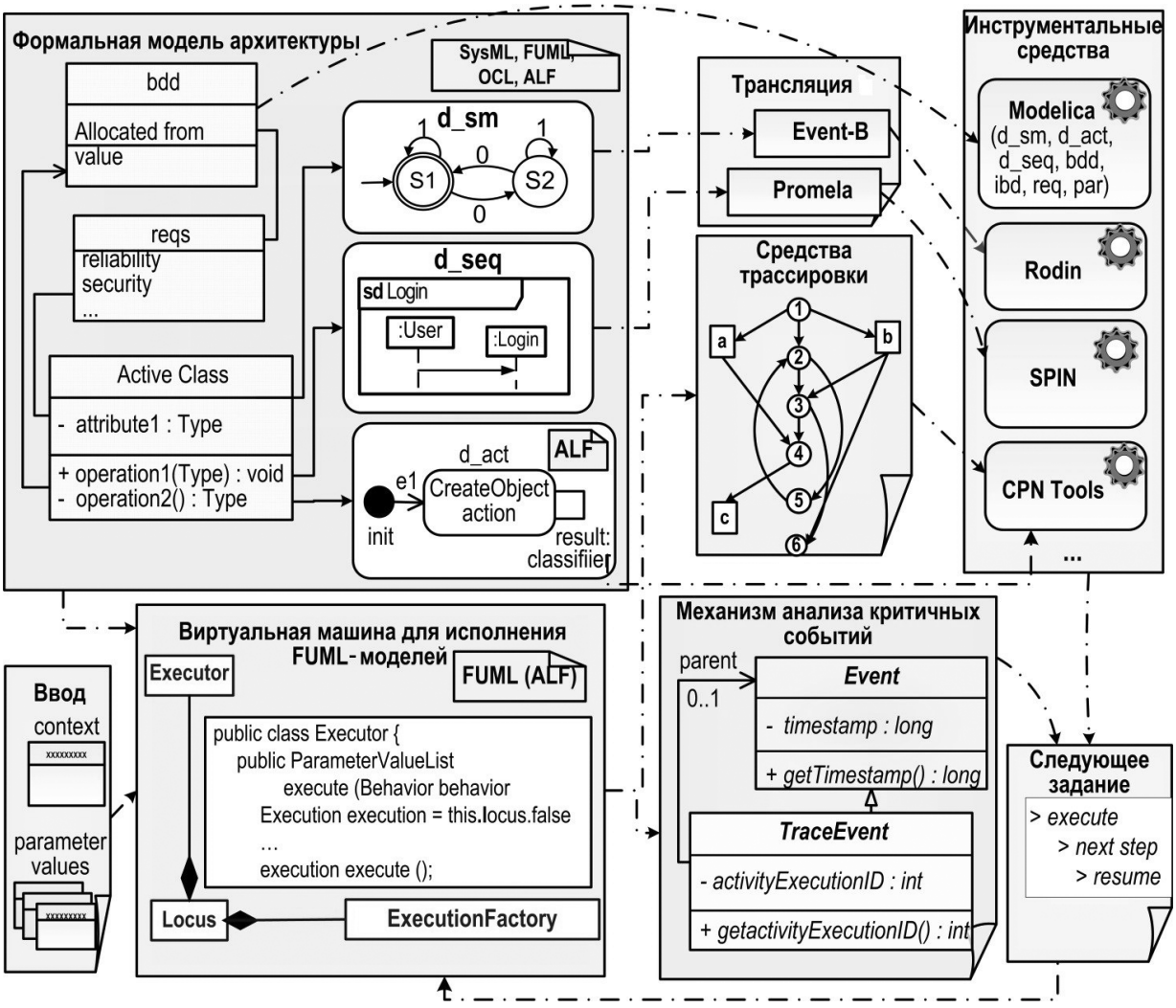

Рис. 2. Состав и структура комплекса программных средств, обеспечивающих верификацию и валидацию проектных решений

ной модели проектных решений в среде виртуальной машины VM FUML. Проверка выполняется посредством трассировки по контрольным точкам и с помощью анализа критичных событий при непосредственном участии экспертов и аналитиков.

Кроме того, модели, представленные в виде диаграмм активностей (d_act), транслируются на язык CPN ML и затем верифицируются средствами CPN Tools. Диаграммы последовательностей (d_seq) транслируются на язык Promela для последующей верификации в среде SPIN. Диаграммы состояний (d_sm) транслируются на язык EventB и проверяются с помощью средств Rodin. Преобразование представленных выше поведенческих диаграмм на входные языки соответствующих средств автоматической верификации выполняется с помощью разработанных для этого программных компонентов единой модельно-языковой и информационно-программной среды.

Для верификации технической части проектных решений предлагается использовать средство моделирования киберфизических систем OpenModelica [21]. Проектные решения, представленные в виде SysML-диаграмм блоков и классов, с помощью специальных программ транслируются в соответствующие конструкции объектно-ориентированного языка Modelica.

В случае обнаружения каких-либо дефектов проектные решения возвращаются архитекторам системы на доработку. Исправленный проект подвергается повторной процедуре валидации и верификации. В результате реализации данного последовательно-итерационного и программно-управляемого процесса будут разработаны проектные решения, в полной мере соответствующие предъявленным к КВИС требованиям пользователей, нормативных документов и условиям применения.

Заключение

Фундамент обеспечения корректного и надежного функционирования КВИС закладывается на этапах обоснования комплекса требований и разработки проектных решений. Для обеспечения требуемого качества этих артефактов процесса разработки КВИС и повышения качества их программно-аппаратной реализации, а также эффективности их последующего применения предлагается разработать и предоставить всем участникам процесса создания КВИС единую модельно-языковую и информационно-программную среду. Для разработки такой среды предложено использовать языки моделирования и проектирования SysML, FUML, OCL, математический аппарат СП, ВА. Для обеспечения автоматизированного построения формальных описаний этих требований в среде визуального моделирования разработаны специальные шаблоны, представляющие собой расширения диаграмм вариантов использования, деятельности, состояний, взаимодействия, блоков для внешнего и внутреннего представлений компонентов системы, активных и пассивных классов. Выполнение автоматизированных процедур верификации и валидации комплекса требований и проектных решений реализуется с помощью разработанного для этого модельного, информационного, алгоритмического и программного обеспечения, а также программных средств верификации и анализа CPN Tools, Rodin и SPIN.

Применение представленного в статье комплекса методов, моделей, алгоритмов и средств позволит реализовать программно-управляемый процесс разработки и сопровождения КВИС, основными достоинствами которого являются:

- реализация автоматизированных процедур верификации, валидации и коррекции комплекса требований и проектных решений как в случае обнаружения каких-либо дефектов, так и при изменении самих требований или условий эксплуатации КВИС, что позволит улучшить экономические показатели в части снижения финансовых и временных затрат, связанных с выполнением дополнительных работ;

- сокращение семантических, логических и терминологических разрывов в представлениях о системе, имеющихся у всех участников процессов создания и применения КВИС, благодаря использованию единой модельно-языковой и информационно-программной среды.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации». – URL: http://www.consultant.ru/document/cons_doc_LAW_220885/ (дата обращения: 25.05.2019).

2. The Standish Group report. – URL: https://www.standishgroup.com/store/services/10-chaos-report-decision-latency-theory-2018-package.html (дата обращения: 21.05.2019).

3. Systems Engineering and Software Engineering. – URL: https://www.sebokwiki.org/wiki/Systems_Engineering_and_Software_Engineering (дата обращения: 25.05.2019).

4. Hart Laura E. Introduction To Model-Based System Engineering (MBSE) and SysML. – URL: https://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf (дата обращения: 21.05.2019).

5. Friedenthal S., Moore A., Steiner R. A Practical Guide to SysML. The Systems Modeling Language. Elsevier Inc., 2012. – 640 p.

6. Delligatti L. SysML Distilled A Brief Guide to the Systems Modeling Language. Pearson Education, Inc., 2014. – 301 p.

7. Advanced and efficient execution trace management for executable domain-specific modeling languages / E. Bousse, T. Mayerhofer, B. Combemale, B. Baudry // Software and Systems Modeling. – URL: https://link.springer.com/ article/10.1007/s10270-017-0598-5 (дата обращения: 27.04.2019).

8. Burgueno L., Gogolla M. Teaching Model Views with UML and OCL. – URL: http://ceur-ws.org/Vol-2019/edusymp_4.pdf (дата обращения: 21.05.2019).

9. Modeling using alf action language. – URL: https://www.conformiq.com/2016/05/modeling-using-alf-action-language/ (дата обращения: 12.06.2019).

10. CPN Tools Homepage – CPN Tools is a tool for editing, simulating, and analyzing Colored Petri nets. – URL: http://cpntools.org/ (дата обращения: 27.05.2019).

11. Rodin. – URL: http://www.event-b.org/ (дата обращения: 11.05.2019).

12. Карпов Ю.Г., Шомшина И.В. Введение в язык Promela и систему комплексной верификации Spin : учеб. пособие. – СПб. : Изд-во Политехнического ун-та, 2010. – 110 с.

13. Messaoud Rahim, Malika Boukala-Ioualalen, Ahmed Hammad. Petri Nets Based Approach for Modular Verification of SysML Requirements on Activity Diagrams. – URL: http:// citeseerx.ist.psu.edu/viewdoc/ (дата обращения: 27.04.2019).

14. Твардовский А.С., Лапутенко А.В. О возможностях автоматного описания параллельной композиции временных автоматов // Труды ИСП РАН. – 2018. – Т. 30, вып. 1. – С. 25–40.

15. Bengtsson J., Wang Yi. Timed Automata: Semantics, Algorithms and Tools. Uppsala University. – URL: https://www.fi.muni.cz/~xpelanek/IA158/TA-intro.pdf (дата обращения: 27.05.2019).

16. Eclipse OCL. – URL: https://projects.eclipse.org/projects/modeling.mdt.ocl (дата обращения: 27.05.2019).

17. Openmodelica. – URL: https://www.openmodelica.org/ (дата обращения: 27.05.2019).

REFERENCES

1. *Federalnyi zakon ot 26.07.2017 No. 187-FZ 'O bezopasnosti kriticheskoi informatsionnoi infrastruktury Rossiiskoi Federatsii'* [Federal Low No. 187-FZ dated July 26, 2017. On the Security of the Critical Information Infrastructure of the Russian Federation]. Available at: http://www.consultant.ru/document/cons_doc_LAW_220885/ (accessed: 25.05.2019).

2. *The Standish Group Report.* Available at: https://www.standishgroup.com/store/services/10-chaos-report-decision-latency-theory-2018-package.html (accessed: 21.05.2019).

3. *Systems Engineering and Software Engineering.* Available at: https://www.sebokwiki.org/wiki/Systems_Engineering_and_Software_Engineering (accessed: 25.05.2019).

4. Hart Laura E. *Introduction to Model-Based System Engineering (MBSE) and SysML.* Available at: https://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf (accessed: 21.05.2019).

5. Friedenthal S., Moore A., Steiner R. *A Practical Guide to SysML. The Systems Modeling Language.* 2012. Elsevier Inc. (Available at: https://www.elsevier.com), 640 p.

6. Delligatti Lenny. *SysML Distilled a Brief Guide to the Systems Modeling Language.* Addison-Wesley Professional PTG Publ., 2014. 301 p.

7. Bousse E., Mayerhofer T., Combemale B., Baudry B. Advanced and Efficient Execution Trace Management for Executable Domain-Specific Modeling Languages. *Software and Systems Modeling.* Available at: https://link.springer.com/article/10.1007/s10270-017-0598-5 (accessed: 27.04.2019).

8. Burgueno L., Gogolla M. *Teaching Model Views with UML and OCL.* Available at: http://ceur-ws.org/Vol-2019/edusymp_4.pdf (accessed: 21.05.2019).

9. *Modeling Using Alf Action Language.* Available at: https://www.conformiq.com/2016/05/modeling-using-alf-action-language/ (accessed: 12.06.2019).

10. *CPN Tools Homepage – CPN Tools is a Tool for Editing, Simulating, and Analyzing Colored Petri Nets.* Available at: http://cpntools.org/ (accessed: 27.05.2019).

11. *Rodin.* Available at: http://www.event-b.org/ (accessed: 11.05.2019).

12. Karpov Iu.G., Shomshina I.V. *Vvedenie v iazyk Promela i sistemu kompleksnoi verifikatsii Spin. Uchebnoe posobie* [Promel Language and Complex Spin-Verification System. Introduction. Textbook]. St. Petersburg, Polytechnic University Publ., 2010. 110 p.

13. Rahim M., Boukala-Ioualalen M., Hammad A. *Petri Nets Based Approach for Modular Verification of SysML Requirements on Activity Diagrams.* Available at: http://citeseerx.ist.psu.edu/viewdoc/ (accessed: 27.04.2019).

14. Tvardovskii A.S., Laputenko A.V. O vozmozhnostiakh avtomatnogo opisaniia parallelnoi kompozitsii vremennykh avtomatov [On the Possibilities of FSM Description of Parallel Composition of Timed Finite State Machines]. *Trudy ISP RAN* [Proceedings of the Institute for System Programming of the RAS], 2018, vol. 30, iss. 1, pp. 25–40.

15. Johan Bengtsson and Wang Yi. *Timed Automata: Semantics, Algorithms and Tools.* Uppsala University. Available at: https://www.fi.muni.cz/~xpelanek/IA158/TA-intro.pdf (accessed: 27.05.2019).

16. *Eclipse OCL.* Available at: https://projects.eclipse.org/projects/modeling.mdt.ocl (accessed: 27.05.2019).

17. *Openmodelica.* Available at: https://www.openmodelica.org/ (accessed: 27.05.2019).