

DOI: 10.15514/ISPRAS-2021-33(6)-19



Моделирование операционных, программных и технических систем в проектах РФФИ

^{1,2} Е.М. Лаврищева, ORCID: 0000-0002-1160-1077 <lavr@ispras.ru>

^{1,3,4} А.К. Петренко, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25

² Московский физико-технический институт,

114701, Россия, Московская область, г. Долгопрудный, Институтский пер., 9

³ Московский государственный университет имени М.В. Ломоносова, 119991, Россия, г. Москва, Ленинские горы, д. 1

⁴ НИУ Высшая школа экономики,

101978, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. Рассмотрен широкий круг вопросов теории и практики разработки крупномасштабного программного обеспечения и гибридных программно-аппаратных систем, в том числе операционных систем. Затрагиваются вопросы управления конфигурацией, моделирования и верификации таких систем, построения онтологических моделей предметных областей, связанных с прикладным и системным программным обеспечением. Такое многостороннее рассмотрение необходимо для обеспечения надежности, безопасности и эволюционного развития в течение многолетнего периода эксплуатации инфраструктурных и критически важных систем. Статья основана на материалах исследований, выполненных в рамках двух проектов РФФИ. Помимо недавних результатов, авторы уделяют внимание истории развития исследований в соответствующих областях в Советском Союзе.

Ключевые слова: моделирование; программирование; повторное использование; вариабельность; безопасность; защита; надежность; качество

Для цитирования: Лаврищева Е.М., Петренко А.К. Моделирование операционных, программных и технических систем в проектах РФФИ. Труды ИСП РАН, том 33, вып. 6, 2021 г., стр. 265-280. DOI: 10.15514/ISPRAS-2021-33(6)-19

Благодарности: Данная работа выполнена при поддержке гранта РФФИ 19-01-00206.

Modeling of operational, software and technical systems in RFBR projects

^{1,2} E.M. Lavrischeva, ORCID: 0000-0002-1160-1077 <lavr@ispras.ru>

^{1,3,4} A.K. Petrenko, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>

¹ Ivannikov Institute for System Programming of the Russian Academy of Sciences,

25, Alexander Solzhenitsyn st., Moscow, 109004, Russia

² Moscow Institute of Physics and Technology (MIPT),

9, Campus per., Dolgoprudny, Moscow region, 114701, Russia

³ Lomonosov Moscow State University,

GSP-1, Leninskie Gory, Moscow, 119991, Russia

⁴ National Research University, Higher School of Economics

20, Myasnitskaya ulitsa, Moscow, 101978, Russia

Abstract. A wide range of issues of theory and practice in the development of large-scale software and hybrid software intensive systems, including operating systems, is considered. Issues of configuration management, modeling and verification of such systems, construction of ontological models of subject areas related to application and system software are touched upon. Such multilateral consideration is necessary to ensure reliability, security and elastic development during the multi-year period of operation of infrastructural and mission-critical systems. The paper is based on the materials of studies carried out within the two RFBR projects. In addition to recent results, the authors pay attention to the history of the development of research in the relevant areas in the Soviet Union.

Keywords: modeling; programming; reuse; variability; security; protection; reliability; quality

For citation: Lavrischeva E.M., Petrenko A.K. Modeling of operational, software and technical systems in RFBR projects. Trudy ISP RAN/Proc. ISP RAS, vol. 33, issue 6, 2021, pp. 265-280 (in Russian). DOI: 10.15514/ISPRAS-2021-33(6)-19

Acknowledgements. This work was supported by the RFBR grant No. 19-01-00206.

1. Введение

Программирование прошло долгий путь с 50-х годов XX века до наших дней; в результате накоплен опыт создания разнообразных системных и прикладных программ, в частности: операционных систем, библиотек математических методов решения физико-технических, прикладных и предметных задач для прикладных областей (физика, биология, медицина, химия, генетики и др.). Широкое применение получили функциональные компоненты многообразного использования (КПИ) в разных областях знаний и метод конвейерной сборки технических средств (устройств, приборов) и робототехнических изделий в биотехнике, медицине, химии и др. Каждый ресурс (модуль, объект, компонент) и системы из них проверялись на безопасность, надежность, качество и защиту обработки данных. Международное сообщество сформировало много стандартов по разработке, оцениванию и управлению созданием сложных систем: АИС, АСНИ, АСУ, АСУТП и др. Созданы и действуют робототехнические устройства для проведения физиологических, кардиологических, травматологических, онкологических исследований и проведения хирургических операций. В университетах, институтах и школах страны сформировались программы обучения студентов современным основам программирования и инженерии разработки разных математических задач, технических и операционных систем.

Одним из основных направлений создания программно-технических систем (ПТС) является теория математического моделирования сложных систем из готовых ресурсов, отражающих сформировавшиеся знания в виде интеллектуальных, информационных, операционных и вычислительных средств. В мировом сообществе в рамках информатизации с 1992 созданы специальные информационные технологические инструменты (E-science, Semantic Web, Grid, Etics и др.), обеспечивающие приобретение и извлечению знаний, накопленных в

разных предметных областях знаний для их использования. Одним из средств концептуализации знаний Semantic Web является онтологический аппарат представления знаний в научных, технических, физических, математических, биологических областях.

Начиная с 2001 года, проводились международные конференции VaMoS (International Working Conference on Variability Modelling of Software-Intensive Systems), сформировавшие новые методы моделирования ОС и моделирования задач технических и программных областей: физических экспериментов, моделирования медицинских, экономических, социальных, финансовых задач. Одним из путей решения таких задач является технология производства программных продуктов (ПП), удовлетворяющая определенным экономическим критериям по организации моделирования, разработки и изготовления продуктов высокого качества и производительности с использованием готовых ресурсов, КПИ со стандартными или унифицированными интерфейсами (CALL, RPC, RMI, IDL, WSDL, API, ABI), способствующих проведению формализованной сборки ресурсов в ЯП 4 поколения (C++, Java, Python, Ruby и др.) на различных программных платформах.

2. Периоды развития средств моделирования и программирования на ЭВМ в СССР

2.1 Начальный период моделирования ОС ПО, программных и технических систем

На первых ЭВМ -МЭСМ (1955), СТРЕЛА (1956), М-20 (1958) были реализованы ОС и ПО для проведения работ по трансляции, отладки, тестированию программ на первых языках программирования ЯП Algol-60, Fortran, PL/I, Cobol и др. Программы вводились с внешних устройств ввода, вывода и ОС управляла их обработкой (трансляция и др.) на ПО (трансляция и др.) с сохранением результатов во внешней памяти ЭВМ. Для трансформации информации в код ЭВМ использовались трансляторы (ТА1 –ТА4) с ЯП, отладчики, тестировщики, сборщики и др. Первыми программам были простые численные задачи с данными для вычисления. Результаты запоминались в библиотеках программ. По решению ГКНТ СССР были созданы Республиканские фонды алгоритмов и программ (РФАП), так называемые склады программ, и разработан метод объединения, сборки отдельных программ в разных ЯП в большие системы для их выполнения в операционной среде ЭВМ. Одновременно программировались разные технические, математические, интеллектуальные и информационные задачи для постановки и решения на ЭВМ.

2.2 Глобальные цели для инициации процессов моделирования технических систем

Понятие *модель* первоначально сформировалось в 1613 году при строительстве зданий в Древней Греции в виде образца здания и модели задачи Аль Хорезми и Ньютона для численных расчетов задач на аналитических машинах. Моделирование техники развивалось при строительстве зданий, мостов, лодок, самолетов и др. Модель отражала сущность технического предмета или явления и связи элементов. В 18-19 веке в России созданы технические средства -радио, телефон, самолет, автомобиль... Значительный скачок развития технических средств и прикладной математики произошел после окончания Великой Отечественной войны (ВОВ) 1945 года по следующим направлениям: – передовые систем вооружений, обусловленные ходом ВОВ; –гонка вооружений между СССР и США; – ядерная программа И.В. Курчатова после атомного взрыва в Хирасимо и Нагасаки; – ракетостроение, аэродинамика, баллистика, космос и др.; –электронная техника, радиосвязь и численное решение уравнений Ньютона, Лейбница, Максвелла и др.; –активная разведка природных ресурсов, добыча нефти и газа в нашей стране.

2.3 Моделирование техники и математики

Важным стимулом к развитию теории практики программирования стали задачи создания технологий военного назначения. Сразу после окончания Второй мировой войны Министерство Радиопромышленности СССР создало программу развития радиотехнических и радиофизических приборов с обеспечением высокого качества и безопасности. В рамках предприятий военно-промышленного комплекса (ВПК) были созданы специализированные устройства и средства радиотехнического, радиолокационного, авиационного, космического назначения. Была создана атомная бомба и космический корабль с Ю. Гагариным борту (1961). Весь мир был поражен уровнем достижений советской науки. Были созданы специализированные ЭВМ (ПРА-6.0, МАПА, АРГОН, АОУ6 и др.), а также радиолокационная техника для наведения и слежения за движением самолетов, подводных лодок, ракет, кораблей и др. Созданы программные комплексы ПРОТВА, ЯУЗА, РУЗА, ПРОМЕТЕЙ для решения задач противовоздушной обороны и других задач оборонной тематики (подробнее см. [1] и другие книги В.В. Липаева).

2.4 Моделирование математических задач энергетики, баллистики и геофизики

Ядерный взрыв в Хирасимо и Нагасаки в 16 июля 1945 года послужил поводом для создания в нашей стране Атомного реактора под руководством И.В. Курчатова и проведения многочисленных расчетов, связанных с распадом ядерного урана и плутония для ядерной бомбы. Была создана теория цепной реакции распада урана и формирования математических методов в ядерной энергетике, что привело к созданию в стране новых энергетических и технических устройств (РДС-3-7), атомной бомбы в 1949 году, первая АТС и атомный ледокол им. Ленина (1950). После развала идет создание Термоядерного реактора у нас и Европейского проекта коллайдера и др. Моделирование методов в энергетике и баллистике активно проводилось на первой МЭСМ, созданной член-корр. С.А. Лебедевым (1948-1951) коллективом лаборатории Института электротехники АН УССР. На МЭСМ под руководством академика М.В. Келдыша и с участием академика А.А. Дородницына решались математические задачи внешней баллистики, электротехнические задачи, задачи теории упругости; обеспечения устойчивости энергосистем, задачи расчета тепловых напряжений строительных конструкций, задачи выбор оптимальных параметров шахтных канатов, геодезические задачи, оценка объемов земляных работ автодорог (и БАМ) и многие другие ([2-4]).

2.5 Моделирование физических задач и механики сплошных сред

Проведение вычислительных экспериментов с помощью математической модели физического эксперимента, включающей характерные свойства физического явления, алгоритма свойств и характеристик этого явления, задаваемых в виде простых формул и операций программы решения задач (математической физики, нелинейной механики и сплошных сред и др.) на ЭВМ. Одними из первых важнейших вычислительных задач была задачи о ядерном взрыве и об обтекании тел сверхзвуковым потоком газа. Исследование физических объектов проводилось путем анализа и выбора схем эксперимента для определения элементов физического устройства. Затем составляется разностная схема и проверка ее устойчивости. При математическом моделировании этих задач создавалась приближенная модель. Разработка рациональных численных моделей проводилась с помощью интеллектуальных гибридных *экспертных систем* для прогнозирования поведения объектов и *распознавания* образов объектов с учетом временных интервалов. Проблема *интеллектуализации* экспертных систем была перенесена на решение задач поиска динамических инвариантов для соответствующих классов эволюционных задач. При моделировании численных задач в механике сплошных сред реализованы методы

дискретных вихрей, Монте-Карло, метод неопределенных коэффициентов стационарных методов и расщепления физических процессов с сильно изменяющимися границами области интегрирования [5, 6].

2.6 Первые проекты системного программирования в СССР

Первыми примерами системных программ были прообразы операционных систем, компиляторов, редакторов внешних связей и загрузчиков, что несколько позже стало называться системами программирования. Среди наиболее значимых результатов следует отметить:

- Диспетчеры и операционные системы для МЭСМ, М-20, Стрела, Весна (М.Р. Шура-Бура, В.С. Штаркман), БЭСМ Д-68 и НД-70 (Л.Н. Королев, А.Н. Томилин, В.П. Иванников и др.),
- ОС ИПМ для М-220 1967 г. ИПМ (И.Б. Задыхайло, С.С. Камынин и др.),
- Монитор -Дубна (Н.Н. Говорун и В.П. Шириков).
- Программирующая программа для языка операторных схем Ляпунова А.А. сделана в 1953 г. в МГУ (Э.З. Любимский, А.П. Ершов) и трансляторы с языка Алгол-60 (ТА) для работы с программами на этом ЯП:
 - ТА1–С.С. Лавров (ЛГУ, 1962); ТА2–М.Р. Шура-Бура и Э.З. Любимский (ИПМ, 1963);
 - ТА3 -(Альфа-система) версия языка Алгол-60 (СО АН СССР, 1964);
 - ТА4 – Е.Л. Ющенко, Е.М. Лаврищева для УВК Днепр-1, Днепр-2 (ИК АН УССР).
- Транслятор с языка АЛМО (Любимский Э.З., 1965).
- Первая Библиотека стандартных математических подпрограмм ИС-2 для М20 в МГУ (Е.А. Жоголев, Г.С. Росляков, Н.П. Трифонов, М.Р. Шура-Бура. Система стандартных подпрограмм) и др.
- Алгоритмизация задач прикладной математики для первых ЭВМ с использованием теории граф-схемного языка А.А. Ляпунова (1955). Описание математических элементов задач с помощью математических и логических операций и операторов перехода ориентированного графа с одной входной и одной выходной вершиной.

2.7 Теория графового моделирования задач техники и математики

Впервые теория графов была открыта Эйлером в 1736 г. и описана в статье про строительство Кенигсбергских мостов. В течение почти ста лет эта теория мало использовалась. Интерес к теории графов возродился в середине XIX в связи с развитием естественных наук (электрических сетей, моделей кристаллов, структур молекул и др.) и математической логики. Решение многих математических головоломок, шахматных задач формулировались в терминах графовой теории. Последние 50 лет ознаменовались использованием теории графов в атомной энергетике, нейронных, нанобиологических, генетических, геофизических, космических задачах и др., например, физический эксперимент Европейского проекта «Коллайдер», ядерный реактор, компьютеры, информатика, телекоммуникация, экономика и др.

Теория графов в программировании: А.С. Ляпунов (МГУ), А.П. Ершов, Касьянов В.И, Иткин В.Э., Евстигнеев А.А. и др. (НГУ) и др. Эта теория активно развивается в компьютерных науках и при моделировании систем. Математический аппарат инцидентности и смежности использовался для доказательства графовых, нейрографовых структур и др. (Апонович, НГУ). Графовые структуры используются при проведении физических экспериментов, разработке информационных и интеллектуальных систем. В ИСП РАН графовая теория применялась при моделировании вариантов ОС Linux (2018-2021) с обеспечением надежности и безопасности. Подробнее с этим направлением можно ознакомиться по работам [7-14].

2.8 Развитие информационных и интеллектуальных систем

Начиная с 2000 годов, в мировой сообществе проводились конференции VAMOS 2002-2021. Многие компании по производству программных продуктов проводили практические работы по созданию ПП и возможности изменения отдельных программ в действующих системах и генерацию вариантов продуктов. Software Engineering Institute of USA разработал концепцию продуктовой линии производства продуктов ПС и СПС – Product Line/ Product Family. Продуктовая линия позволяет собирать разные варианты систем из готовых программных элементов (программ, компонентов, assets и др.) под требования потребителей. Создавались варианты ПС и СПС небольших размеров с требуемым количеством функций. В результате сформировался метод сборки вариабельных продуктов и систем разного назначения. Основу метода составляла модель внешних характеристик MF (Feature Model) функций с заданием интерфейсов связи с другими функциями системы. Появился стандарт конфигурационной сборки IEEE 828: Configuration-2009 функциональных элементов в структуру системы. Стандарт позволял манипулировать характеристиками системы для создания разных вариантов выпускаемых при производстве систем разного назначения.

2.9 Вариабельность систем

Модель вариабельности MF включает оценочную модель для сбора и накопления ситуаций, возникающих при разработке и сопровождении семейств СПС, документирования этих ситуаций в терминах значений уровней изменчивости и степени соответствия системы потребностям предметной области. Для управления вариабельностью используется фреймворк COVAMOF (ConIPF Variability Modeling Framework), который погружается в среду MS VisualStudio моделирует создание ПС, СПС по модели MF и архитектуре из готовых компонентов (рис. 1). Основные публикации: [15-20].

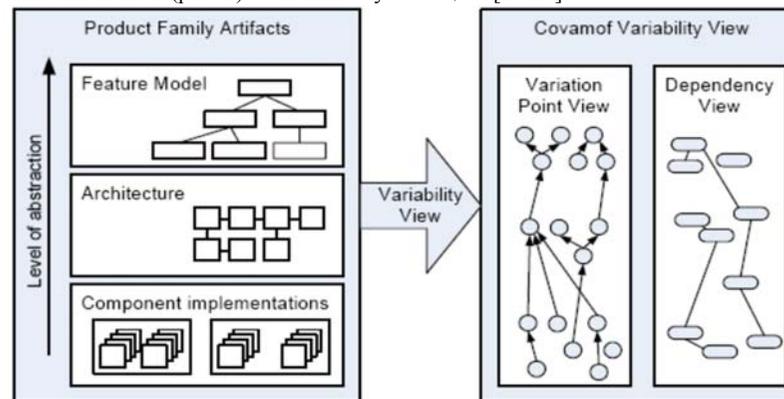


Рис. 1 Обработка на вариабельность моделей и компонентов
Fig. 1 Processing for variability of models and components

2.10 Моделирование онтологии систем в Семантик Веб

В Семантик Веб для поддержки онтологии имеются инструменты: Protege, Eclipse, основанные на фреймворке представления знаний и редактирования моделей в UML, XML, SHOE, DAML, OML, DSL, RDF и RDFS. В Protege имеются готовые плагины: Protege OWL Plugin подобный тестам JUnit, средства генерации описаний в языке Java и возможность преобразовывать в формат XML-схемы на платформе Eclipse в XML Schema Infoset Model (XSD). Онтология приложения, ПС, СПС описывается с помощью классов, слотов, фасетов и аксиом. Классы содержат описание базовых понятий предметной области, слоты

определяют свойства понятий, Фасеты описывают свойства слотов (типы и диапазоны значений). Аксиомы определяют ограничения (правила) и отношения. Классы могут быть абстрактными или конкретными. Абстрактные классы являются контейнерами конкретных классов и могут содержать абстрактные атрибуты. Конкретные классы содержат слоты, которыми могут быть значения атрибутов. Фасеты и слоты задаются графовыми XML-схемами. Кардинальный слот определяет возможное количество значений слота, ограничение типа значений (например, целое, буквенное и др.), предельные значения (мин. и макс.) для числовых слотов и т.п. Средствами онтологии описывается структура домена ЖЦ стандарта ISO/IEC 12207 с выходным результатом в виде графа XML-структуры.

2.11 Онтология ЖЦ ISO/IEC 12207-2012

Стандарт ISO/IEC ЖЦ 12207–2012 является основным инструментом планомерного процессного изготовления ПС. Он включает три категории процессов: 1) основные процессы; 2) процессы поддержки; 3) организационные процессы. Эти процессы включают способы и форм представления и выполнения результатов (рис.2, 3).



Рис.2. Схема основных процессов ЖЦ ПС для онтологии
Fig.2. Scheme of the main processes of the life cycle of PS for ontology



Рис. 3. Схема вспомогательных процессов ЖЦ ПС
Fig 3. Scheme of auxiliary processes of the life cycle of the PS

Стандарт жизненного цикла содержит в себя также вспомогательные процессы, которые регламентируют дополнительные действия при проверке продукта, управлении проектом и качеством. Как правило, в зависимости от целей конкретного проекта главный разработчик и менеджер проекта выбирают процессы, действия и задачи, выстраивая определенную модель жизненного цикла для этого проекта. Данный стандарт (в варианте 2007 года) включает в себя (табл. 1) 17 процессов, 74 подпроцесса и 232 технологические операционные задачи (действия). Их необходимо и достаточно для проектирования систем с помощью процессного

подхода. Некоторые системные фирмы производители программных пакетов реализуют отдельные фрагменты или варианты этого стандарта.

Табл. 1. Процессы, подпроцессы и задачи жизненного цикла
Table 1. Processes, sub-processes and life cycle tasks

Классы	Процесс	Действие	Задача
Основные процессы	5	35	135
Процессы поддержки	8	25	70
Организационные процессы	4	14	27
Всего	17	74	232

Концепция автоматизации стандарта жизненного цикла средствами онтологии является новой. Базовыми понятиями онтологии являются процессы жизненного цикла, их взаимосвязи по передаваемым результатам проектирования программных систем, а также функции процессов для их реализации и выполнения. Если все 17 процессов будут реализованы, то можно из них генерировать некоторые подмножества жизненного цикла, как вариант рабочего жизненного цикла для конкретного применения. Нами рассмотрена онтология процесса тестирования.

Для проектирования онтологии жизненного цикла могут использоваться языки BPMN и VPEL. Жизненный цикл представляется с помощью словарей понятий, концептов и отношений между ними в среде Protégé. Полученное онтологическое описание жизненного цикла трансформируется на язык XML, который при реализации размечает данные домена жизненного цикла и устанавливает связи и обмен данными между процессами.

Домен ЖЦ занимает центральное место в программной инженерии, основным назначением которого является поддержка методов и средств изготовления сложных программных систем. Изучающие эти методы и средства, а также современные стандарты ЖЦ ISO/IEC 12207–2007 и ISO/IEC 11404–2007 GPD (General Purpose Datatypes) легли в основу разработки экспериментального варианта онтологии ЖЦ. Концепция онтологии ЖЦ обсуждалась в КНУ на научных семинарах кафедры теории программирования и технологии, кафедры информационных систем и в МФТИ на кафедре информатика и прикладная математика. Приводимое описание онтологии получено с участием студентов.

2.12 Моделирование онтологии «Вычислительная геометрия» в Семантик Веб

Задачи вычислительной геометрии являются классическими, поэтому эту область можно использовать как хороший пример демонстрации построения онтологий. К основным понятиям вычислительной геометрии относятся:

- выпуклая оболочка с набором точек, необходимых для нахождения наименьшего выпуклого многоугольника, содержащего все точки;
- пересечение отрезков в наборе отрезков;
- триангуляция Делона;
- диаграмма Вороного, содержащая набор точек, разделяющих пространство на сектора, каждая точка которых ближе к набору других;
- задача ближайшей пары точек из набора точек для нахождения кратчайшего пути;
- евклидов кратчайший путь, который соединяет две точки в Евклидовом пространстве (с полигональными препятствиями) кратчайшим образом;
- триангуляция многоугольника и разбивка на внутренние треугольники;
- метод Грехема и Джарвиса для обхода точек в простых и выпуклых многоугольниках;
- метод быстрой сортировки на множестве точек и линий пересечения фигур;
- метод М.Шеймоса организации данных для выпуклых многоугольников;

– графовые структуры Штейнера для решения задач триангуляции и поиска соседа.

Рассматриваются статические и динамические задачи. К статическим задачам относятся:

- *выпуклая оболочка*: имея набор точек необходимо найти наименьший выпуклый многоугольник, который содержит все точки;
- *пересечение отрезков*: найти все пересечения в наборе отрезков;
- *триангуляция Делона*;
- *диаграмма Вороного*: имеется множество точек на плоскости, требуется разделить это множество на области, чтобы каждая такая область образовывала множество точек, более близких к одному из элементов исходного множества, чем к любому другому его элементу;
- *задача ближайшей пары точек*: имея набор точек найти кратчайшее расстояние;
- *евклидов кратчайший путь*: соединить две точки Евклидова пространства (с полигональными препятствиями) кратчайшим образом.
- *триангуляция многоугольника*: имея многоугольник, разбить его на внутренние треугольники.

Задачи геометрического поиска включают пространство поиска:

- *региональный поиск*: обработать набор точек, с целью эффективного поиска набора точек, содержащихся в заданном регионе.
- *локализация точки*: имея разбиение пространства на регионы, создать структуру данных, которая позволит эффективно определить, в каком регионе находится данная точка.
- *поиск ближайшего соседа*: обработать набор точек чтобы иметь возможность эффективно найти точки, близкие к заданной.
- *трассировка лучей*: для заданного набора объектов в пространстве создать структуру данных, которая позволит эффективно определять объекты, пересекающие заданный луч.

Комбинаторная вычислительная геометрия в терминах базовых геометрических объектов: точки, отрезки, многоугольники, многогранники. Численная вычислительная геометрия или машинная геометрия.

2.13 Метод сборки информационных и интеллектуальных ресурсов в ПС и СПС

В начале 70-х годов сформировалось понятие «модуля» как главного элементарного, программного элемента, выполняющего некоторую отдельную функцию в общей архитектуре программной системы. Модуль описывался в ЯП (Algol-60, Fortran, Cobol, Smalltalk) и переводился в код ЭВМ отечественными трансляторами в среде ОС ЭВМ. Модули образуют элементы систем и накапливаются в репозитории ОС ПО. Академик В.М.Глушков высказал идею о том, что программы из модулей будут собираться конвейерным способом, как автомобили из готовых деталей на фабрике Форда. Реализован метод сборки модулей на основе межмодульного интерфейса (1976–1982) в системе АПРОП*. Разработан межмодульный, межязыковый и технологический интерфейс. Получена грамота за интерфейс на конференции «Интерфейс СЭВ (1987) и он стал базовым понятием в технологии программирования.

Метод сборки и библиотека интерфейса (64 примитивов) реализованы в ОС ЕС/IBM360, внедрен в ВПК (В.В. Липаев, МНИИПА) для создания комплексов РУЗА, ПРОМЕТЕЙ, ЯУЗА и специализированных ЭВМ (ПРА-6.0, МАПА, АРГОН, АОУ6 и др.). Система АПРОП была передана в 52 организации СССР и отмечена Государственной премией Кабинета министров СССР (1985). Основные публикации: [18-20].

2.14 Теория сборочного программирования

Сущность теории решения задачи сборки пар разнородных модулей или объектов состоит в построении взаимно однозначного соответствия между множеством формальных и фактических параметров [18].

$V = \{v^1, v^2, \dots, v^k\}$ вызов объекта с множеством формальных параметров;

$F = \{f^1, f^2, \dots, f^{k^l}\}$ вызов объекта и их отображение с помощью алгебраических систем [2–4].

Каждому типу данных T_d^l языка l_a ставится в соответствие алгебраическая система вида:

$$G_{\alpha^t} = \langle X_{\alpha^t}, G_{\alpha^t} \rangle,$$

где X_{α^t} – множество значений рассматриваемого типа данных (ТД), а G_{α^t} – множество операций над объектами данного типа. Операциям преобразования ТД соответствует изоморфное отображение алгебраических систем G_{α^t} в G_{β^d} .

В классе алгебраических систем

$$\Sigma = \{G_{\alpha^b}, G_{\alpha^c}, G_{\alpha^i}, G_{\alpha^r}, G_{\alpha^a}, G_{\alpha^z}\},$$

где тип t – это b – boolean, c – character, i – integer, r – real, a – array, z – record и др. В системе Σ реализованы все виды преобразований для представленных ТД и проведено доказательство изоморфизма алгебраических систем [19] и его отсутствие для множеств неэквивалентных значений X_{α^t} и X_{β^d} . Установлено, что мощности алгебраических систем равны

$$|G_{\alpha^t}| = |G_{\beta^d}|.$$

В парадигме сборки реализованы универсальные формальные основы преобразования типов входных / выходных данных (простых и сложных) FDT к общим типам данных GPD стандарта ISO/IEC 11404 (примитивные, агрегатные, генерированные и неструктурированные) [1–10]. Ниже рассматриваются общие вопросы преобразования FDT, GPD и $GPD \leftrightarrow FDT$, как очень важные для сборки программ для среды Интернет.

Теории объектно-компонентного моделирования посвящены следующие публикации: [18–20].

2.15 Генерация ТД стандарта GPD

Разработана схема генерации ТД GPD в структуры фундаментальных ТД ЯП. Предлагается разработать библиотеку функций генерации ТД стандарта GPD, элементы которой для всех новых ТД этого стандарта выполняют следующие виды операций:

- преобразование ТД, содержащихся в ЯП ($ЯП_1, \dots, ЯП_n$) и входящих в состав ФДТ ТД;
- функции трансформации сложных ТД стандарта GPD, включенных в стандартную библиотеку CTS VS.Net и используемых трансляторами с ЯП этой системы для преобразования сложных ТД к более простым;
- операции взаимодействия компонентов повторного использования, записанных в разных ЯП, интерфейс которых задается в языке IDL.

Все ТД стандарта GPD представлены в виде следующих классов алгебраических систем:

$$\Sigma 1 = \{G_{\alpha^b}, G_{\alpha^c}, G_{\alpha^i}, G_{\alpha^r}\},$$

$$\Sigma 2 = \{G_{\alpha^a}, G_{\alpha^z}, G_{\alpha^u}, G_{\alpha^e}\},$$

$$\Sigma 3 = \{G^s, G^{ns}\},$$

где $G_{\alpha^t} = \langle X_{\alpha^t}, \Omega_{\alpha^t} \rangle$, t – ТД языков L , X_{α^t} – множество значений ТД; Ω_{α^t} – множество операций над ТД; $\Sigma 1$ – простые ТД: $t = b$ (bool), c (char), i (int), r (real); $\Sigma 2$ – сложные (структурные) типы данных: $t = a$ (array), z (record), u (union), e (enum); как комбинации простых ТД; $\Sigma 3$ – сложные, неструктурированные ТД (портфель, контейнер, протокол и т.п.) и сгенерированные из них более простые данные.

В каждом классе этих систем преобразование $t \rightarrow Tq$ для пары языков lt и lq основано на таких свойствах отображений:

- 1) системы $G\alpha^t$ и $G\beta^q$ – изоморфны, если их q, t определены на том же множестве ТД;
- 2) между значениями $X\alpha^t$ и $X\beta^q$ типов данных t, q существует изоморфизм, если множество операций $\Omega\alpha^t$ и $\Omega\beta^q$ разные;
- 3) если множество $\Omega = \Omega\alpha^t \cap \Omega\beta^q$ не пустое, то имеет место изоморфизм двух систем;
- 4) $G\alpha^t = \langle X\alpha^t, \Omega \rangle$ и $G\beta^q = \langle X\beta^q, \Omega \rangle$;
- 5) если типы данных отличаются, например, t – строка, а тип q – вещественное, то между множествами $X\alpha^t$ и $X\beta^q$ не существует изоморфного соответствия.

Отображения сохраняют линейный порядок элементов к виду линейной упорядоченности элементов алгебраических систем из этих классов.

2.16 Теория технологии программирования (ТП)

Теорию ТП как математическую дисциплину сформулировали А.П. Ершов и зарубежные ученые (E. Dijkstra, A. Hoag, Z. Manna, D. Gries, D. Vjotner, K. Parnas др.). Так в докладе на Всесоюзной конференции «Технология программирования» (1986) А.П. Ершов определил:

Теория ТП – это методы синтеза, сборки и конкретизации.

- *Синтезирующее программирование* базируется на методе доказательного рассуждения о правильности программы.
- *Сборочное программирование* осуществляет построение программы из уже существующих (проверенных на правильность) готовых фрагментов программ (reuses) и сборку их в сложную структуру.
- *Конкретизирующее программирование* обеспечивает построение системы по универсальной модели для некоторой предметной области.

Программные элементы требуется верифицировать, тестировать для поиска ошибок и их исправления, а затем после интеграционного тестирования оценивать на качество (см. [21, 22]).

2.17 Методы и средства обеспечения надежности и качества технических, системных и программных средств

В рамках проекта РФФИ 19-01-00206 решались задачи моделирования качественных прикладных, операционных систем и сборки экспериментального варианта ядра OS Linux для применения в комплексах, технике и приборах для областей знаний. Для решения задач сборки варианта OS Linux был проведен анализ свойств используемых приборов с позиций обеспечения надежного функционирования элементов ОС с помощью операций сборки (build, assembly, config, make, waver) функций OS Linux в новое экспериментальное ядро для областей знаний. Операции сборки реализованы в современных операционных средах: make в BSD и GNU; build, assembly в JavaEE, Grid, Etics, IBMSphere, VS.MS, Intel, Unix, Linux и др. Результаты исследований и реализации задач моделирования прикладных и операционных систем с обеспечением качества представлены в работах:

Технология сборки экспериментального варианта ядра OS Linux с обеспечением качества включает следующие процессы.

- 1) Препроцессирование – предварительная обработка функции ядра ОС, компиляция для получения кода версии с поиском ошибок.
- 2) Компиляция формально описанных функций для получения выходного кода с проверкой ошибок в синтаксисе их описания.
- 3) Извлечение функциональных элементов ядра и представления их в классы эквивалентности с проверкой, тестированием MC/DC.

- 4) Связывание (сборка) выделенных элементов ядра с помощью операторов make, config и интерфейса связи элементов и обмена данных между ними из библиотек Интернет.
- 5) Конфигурация структуры с помощью операции Kconfig для получения файла конфигурации системы.
- 6) Тестирование используемых элементов и собранного из них варианта ОС с поиском ошибок с помощью LDV и CPAChecker и оценка правильности компонентов ОС.
- 7) Оценивание элементов сконфигурированного варианта ядра ОС на надежность, безопасность и защиту информации.

Стандартные средства обеспечения безопасности ОС:

- криптография (Cryptographic Application Program Interface – CAPI), включающая контроль ключей, информации и доступа к данным;
- защита данных в системах общего назначения, в состав которых входит система безопасности, ориентированная на противодействие угрозам безопасности и защиты информации в компьютерной сети (КС). Система безопасности обеспечивает:
- санкционированный доступ к объектам КС;
- успешное отражения угроз (внешних или внутренних) в сети;
- контроль прохождения запросов в сети и оценки эффективности безопасности и др.

Модель системы безопасности имеет следующие характеристики:

- авторизация (АВТ), заключающаяся в типизации пользователей или их групп для получения разрешения на совместное использование отдельных ресурсов ПТС;
- аутентификация (АУТ), устанавливающая подлинность личности, посылающей сообщение в сеть для доступа к ресурсу;
- парольная защита (ПЗ) и контроль доступа к информации, программам и устройствам.

Для оценки надежности экспериментальной версии ядра OS Linux использовался метод вероятностной надежности Маркова и методы оценки полноты верификации. Результаты этого направления были опубликованы в следующих работах: [23-26].

В рамках проекта РФФИ №19-01-00206 проведены исследования гарантоспособности с учетом возникающих событий типа аварий и контратак в экспериментальном варианте OS Linux, а также в созданных технических средствах для авиации, промышленности, морского флота и др.

Таким образом, в рамках работ по проекту РФФИ №19-01-00206 сконфигурирован экспериментальный вариант OS Linux для применения в предметных областях знаний (медицина, физика, математика и др.). Созданный вариант ОС проверен на правильность методами верификации, валидации, тестирования и оценки показателей надежности, качества и безопасности. Данный вариант ядра ОС может использоваться для управления процессами решения функциональных задач предметных областей знаний (медицина, физика, биологии и др.).

3. Заключение

Моделирование и программирование технических и математических задач прошли долгий путь развития и сыграли важную роль в истории моделирования и программирования технических, программных, операционных и системных комплексов различного назначения на ЭВМ (1945-2000) в рамках работ гражданской и оборонной тематики.

Созданные и используемые технологии не были стандартизированы у нас в стране, и после 1992 года стали использоваться в основном зарубежные технологии. Это UML (Unified Modeling Language), OWL (Ontology Web Language), WSDL, XML, XSL, RDF, Семантик Веб, WWW3C.

С 2004 года, сформировалась модель (Model Variability) на линиях производства программ Product Line, в Grid-системах и применяется на фабриках программ AppFab. Особое развитие

получил научный сервис в среде Семантик Веб Интернета (www.semantic_web.org), позволяющий производить композитную сборку сервисов (SOA, SCA). Разработаны современные технологии системного программирования, интеллектуальные средства (Data Mining) для анализа данных в Legacy систем, извлечения знаний (Mining knowledge) и данных в разных системах Big Data, Cloud Computing Сети Интернет.

В отечественной практике разработана теория ОКМ моделирования систем и средств преобразования ТД стандарта ISO/IEC 11404 GPD и неструктурных ТД Big Data. Отработана конфигурационная сборка информационных, программных и операционных ресурсов в систему и представлен пример сборки экспериментального ядра OS Linux.

Сборочный процесс производства ИП основывается на контроле деятельности исполнителей и оценке показателей качества как отдельных КПИ, так и вариантов продукта (например, экспериментального варианта OS Linux). В мировой практике для этих целей используются следующие технологии программирования и моделирования:

- конвейерная сборка с планированием трудозатрат, учетом, контролем результатов и др.) в автомобильной, авиационной промышленности и др.;
- анализ смысла решаемых задач в предметных, п сложных прикладных областях и формализованное их описание в виде КПИ для многократного использования при сборке сложных ПТС;
- развитие средств моделирования с обеспечением контроля изменчивости (вариабельности), безопасности, защиты и качества;
- технологии E-science, Semantic Web, Grid для проведения физических экспериментов и создания информационных и интеллектуальных областей знаний и решения соответствующих задач;
- совершенствование методов моделирования сложных систем (ООР, UML, MDA и др.).
- стандарты конфигурационной сборки (IEEE 828 -1996: Configuration; IEEE 828-1996: Configuration; ISO 14598:2000: планирование и управление электроникой; IEC 61508-2000: функциональная безопасность и др.)

Наиболее быстрый переход к такой организации работ состоял в систематизации и формализации операций сборки для многократного использование готовых информационных ресурсов, в том числе с нестандартными типами данных в случае Big Data в Cloud Computing, в современной среде Интернет.

Проблематика моделирования переменных сложных систем начала исследоваться в ИСП РАН в 2014 году под руководством академика В.П. Иванникова. Методы моделирования ПТС с обеспечением качества, защиты и безопасности были представлены в проектах РФФИ №16-01-00352 «Теория и методы разработки переменных программных и операционных систем» (от 24.09.2016) и РФФИ №19-01-00206 «Модели, методы и средства обеспечения надежности программных и технических систем» от 21.02.2020.

Проект РФФИ №16-01-00352-2016 успешно завершен в 2018. В нем принимали участие сотрудники ИСП РАН - Петренко А.К., Карпов Л.Е., Пакулин Н.И., Кулямин В.В., Митулин В.С., Рыжов А.Г.

В проекте РФФИ №19-01-00206 зафиксированы наработки в 2019 - 2021 годов. В проекте принимали участие сотрудники ИСП РАН – Петренко А.К., Митулин В.С., Рыжов А.Г., Зеленев С.В. и др. Проведено исследование и обработка отдельных методов надежности применительно к специальным устройствам и программам (2019-2021). Исследован набор международных методов по проблематике безопасности и защите данных программ и устройств от аварий, угроз и контратак. Эти методы и средства обеспечения безопасности функционирования систем прошли апробацию на отдельных технических, программных и операционных средствах. Методы надежности, качества с обеспечением безопасности и защиты данных описаны и внедрены на задаче создания экспериментального ядра OS Linux.

В исследованиях и в разработке данных проектов принимали участие более 10 ведущих специалистов ИСП РАН отдела «Технологий программирования», а также студенты МФТИ, которые на практике обрабатывали отдельные аспекты теории моделирования сложных систем в 5 магистерских диссертациях. Научные аспекты моделирования систем и обеспечения надежности, безопасности, защиты данных отражены в 25 докладах и статьях на Всероссийских и международных конференциях и в Трудях ИСП РАН.

Результаты исследований проектов РФФИ:

- верификация и тестирование готовых ресурсов (КПИ), используемых при моделировании программных, операционных и технических средств;
- сборка (конфигурация) готовых ресурсов в сложные системы с доказательством их правильности и представления конфигурационной структурой прикладных и OS Linux с помощью операций config, assembly, make, smake, building;
- отработан метод преобразования разнородных типов данных и систем общих типов данных стандарта ISO/IEC 11404 GPD -1996; GPD-2012 для неструктурированных типов данных Big Data и Cloud Computing средствами SOA, SCA и системных средств Интернет;
- исследованы логические спецификации системных и сервисно-компонентных моделей SOA, SCA, SCM WWW3C с использованием новых функций систем;
- реализован метод сборки информационных и операционных ресурсов с использованием новых операций конфигурирования: config, assembly в Grid; build, waver в Etics: make в BSD, smake в GNU и JavaEE. С помощью операций make создан экспериментальный вариант ядра OS Linux и Веб-сайта с использованием сервисных ресурсов SOA, SCA, SOAP WWW3C Интернет;
- отработаны методы обеспечения безопасности, защиты данных, надежности и качества ПТС в клиент-серверной среде Интернет на программных структурах систем для применения на сайте ИСП РАН;
- отработан экспериментальный вариант ядра OS Linux, верифицирован, тестирован и проверен на безопасность, защиту данных, борьбу с атаками и отказами и оценкой качественных показателей – функциональность и отказоустойчивость.

Заключительные работы опубликованы в работах [27-29].

Список литературы / References

- [1] Липаев В.В. Фрагменты истории развития отечественного программирования для специализированных ЭВМ в 50–80-е годы. Синтез, 2003. 126 стр. / Lipaev V.V. Fragments of the history of the development of domestic programming for specialized computers in the 50–80s. Sintez, 2003. 126 p. (in Russian).
- [2] Кикоин И.К. Игорь Васильевич Курчатов. Атомная энергия, том 14, вып. 1, 1963 г., стр. 5-9 / Kikoin I.K. Igor Vasilievich Kurchatov. Atomic Energy, Volume 14, no. 1, 1963, pp. 5-9 (in Russian).
- [3] Марчук Г.И. Численные методы расчета атомных реакторов. Атомиздат, 1958 г., 381 стр. / Marchuk G.I. Numerical methods for calculating nuclear reactors. Atomizdat, 1958, 381 p. (in Russian).
- [4] Келдыш М. В. Избранные труды. Общие вопросы развития науки. Наука, 1985 г., 704 стр. / Keldysh M. V. Selected Works. General issues of the development of science. Science, 1985, 704 p.
- [5] Тихонов А.Н., Самарский А.А. Об однородных разностных схемах. Журнал вычислительной математики и математической физики, том. 1, вып. 1, 1961 г., стр. 5-63 / Tikhonov A.N., Samarskii A.A. Homogeneous difference schemes. Computational Mathematics and Mathematical Physics, vol. 1, issue 1, 1962, pp. 5-67.
- [6] Белоцерковский О.М. Численное моделирование в механике сплошных сред. Наука, 1994 г., 441 стр. / Belotserkovskii O. M. Numerical modeling in continuum mechanics, Nauka, 1994, 441 p (in Russian).
- [7] Апонович З.В. Визуализация больших графов и матрица смежности. Труды XX Всероссийской научной конференции «Научный сервис в сети Интернет», 2018 г., стр. 28-41 / Aponovich Z.V.

- Visualization of large graphs and adjacency matrix. In Proc. of the XX All-Russian Scientific Conference on Scientific Service on the Internet, 2018, pp. 28-41 (in Russian).
- [8] Лаврищева Е.М., Рыжов А.Г. Подход к моделированию систем и сайтов из готовых ресурсов. Труды XX Всероссийской конференции «Научный сервис в сети Интернет», 2018 г., стр. 321-345 / Lavrischeva E.M., Ryzhov A.G. Approach to the modeling of systems and sites from ready resources. In Proc. of the XX All-Russian Conference on Scientific Services on the Internet, 2018, pp. 321-345 (in Russian).
- [9] Лаврищева Е.М., Петренко А.К. Технология сборки интеллектуальных и информационных ресурсов Интернет. Труды XXI Всероссийской конференции «Научный сервис в сети Интернет», 2019 г., стр. 469-488 / Lavrischeva E.M., Petrenko A.K. Technology of Assembly of intellectual and information resources of the Internet, In Proc. of the XXI All-Russian Conference on Scientific Services on the Internet, 2019, pp. 469-488 (in Russian)
- [10] Лаврищева Е.М. Теория графового моделирования сложных систем из модульных элементов для прикладных областей. Austria-science, no. 28, часть 1, 2019 г., стр. 12-30 / Lavrischeva E.M. The theory of graph modeling of complex systems constructed from modular elements for applied areas. Austria-science, no. 28, part 1, 2019, pp. 12-30 (in Russian).
- [11] Lavrischeva E.M., Petrenko A.K., Pozin B.A. Technology of Assembly of Intellectual and Information Resources Internet. In Proc. of the XXI All-Russian Conference on Scientific Services on the Internet, CEUR-WS, vol. 2543, 2019, pp. 247-271.
- [12] Lavrischeva E. The theory graph modeling systems from quality modules of the application areas. In Proc. of th6th International Conference on Actual Problems of System and Software Engineering, CEUR-WS, vol. 2514, 2019, pp. 235-247.
- [13] Лаврищева Е.М., Петров И.Б. Моделирование технических средств и задач прикладной математики на ЭВМ. Труды ИСП РАН, том 32, вып. 6, 2020 г., стр. 167-182 / Lavrischeva E.M., Petrov I.B. Modeling of technical means and problems of applied mathematics on computers. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 6, 2020, pp. 167-182 (in Russian). DOI: 10.15514/ISPRAS-2020-32(6)-13.
- [14] Лаврищева Е.М., Зеленев С.В. Модельный подход к обеспечению безопасности и надежности Web-сервисов. Труды ИСП РАН, том 32, вып. 5, 2020 г., стр. 153-166 / Lavrischeva E.M., Zelenov S.V. Model-Based Approach to Ensuring Reliability and Security of Web-services. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 5, 2020, pp. 153-166 (in Russian). DOI: 10.15514/ISPRAS-2020-32(5)-12.
- [15] Глушков В.М., Лаврищева Е.М. и др. Система автоматизации производства программ (АПРОП). Киев, ИК АН УССР, 1976 г., 134 стр. / Glushkov V.M., Lavrischeva E.M. et al. Program production automation system (APROP). Kiev, IK AN UkrSSR, 1976, 134 p.
- [16] Е.М. Лаврищева, В.Н. Грищенко. Связь разноязыковых модулей в ОС ЕС. М., Финансы и статистика, 1982 г., 137 стр. / E.M. Lavrischeva, V.N. Grishchenko. Linking multilingual modules in the EU OS. M., Finance and Statistics, 1982, 137 p. (in Russian).
- [17] Липаев В.В., Позин Б.А., Штрих А.А. Технология сборочного программирования. М., Радио и связь, 1992 г., 287 стр. / Lipaev V.V., Pozin B.A., Shtrikh A.A. Assembly programming technology. M., Radio and communication, 1992, 287 p. (in Russian).
- [18] Лаврищева Е.М. Теория объектно-компонентного моделирования программных систем. Препринт ИСП РАН, no. 29, 2016 г., 52 стр. / Lavrischeva E.M. The theory of object-component modeling of software systems. ISP RAS Preprint, no. 29, 2016, 52 p. (in Russian).
- [19] Lavrischeva E. Assembling Paradigms of Programming in Software Engineering. Journal of Software Engineering and Applications, vol. 9, issue 6, 2016, pp. 296-317.
- [20] E.M. Lavrischeva. Assembling Paradigms of Programming in Software Engineering. Journal of Software Engineering and Applications, 9, 296-317.
- [21] Ершов А.П. Научные основы доказательного программирования. Вестник АН СССР, no. 10, 1984 г., стр. 9-19 / Ershov A.P. Scientific foundations of evidence-based programming. Bulletin of the Academy of Sciences of the USSR, no. 10, 1984, pp. 9-19 (in Russian).
- [22] Ершов А.П. Отношения методологии и технологии программирования. Информационные материалы и тезисы пленарных докладов II Всесоюзной конференции "Технология программирования", 1986, стр. 10-13/ Ershov A.P. Relations between methodology and technology of programming. Information materials and abstracts of plenary reports of the II All-Union Conference on Programming Technology, 1986, pp. 10-13 (in Russian).
- [23] Лаврищева Е.М., Пакулин Н.В. Методы и средства обеспечения надежности технических и программных средств. Труды 5-й Научно-практической конференции OS DAY, 2018 г., URL:

- http://0x1.tv/20180517F / Lavrischeva E.M., Pakulin N.V. Methods and means of ensuring the reliability of hardware and software. In Proc. of the 5th Scientific and Practical Conference OS DAY, 2018, URL: http://0x1.tv/20180517F.
- [24] Лаврищева Е.М., Пакулин Н.В. и др. Анализ методов оценки надежности оборудования и систем. Практика применения методов. Труды ИСП РАН, том 30, вып. 3, 2018 г., стр. 99-120 / Lavrischeva E.M., Pakulin N.V. et al. Analysis of methods for assessing the reliability of equipment and systems. Practice of methods. Trudy ISP RAN/Proc. ISP RAS, том 30, вып. 3, 2018 г., стр. 99-120 (in Russian). DOI: 10.15514/ISPRAS-2018-30(3)-8.
- [25] Лаврищева Е.М., Зеленев С.В. Модельный подход к обеспечению безопасности и надежности Web-сервисов. Труды ИСП РАН, том 32, вып. 5, 2020 г., стр. 153-166 / Lavrischeva E.M., Zelenov S.V. Model-Based Approach to Ensuring Reliability and Security of Web-services. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 5, 2020, pp. 153-166 (in Russian). DOI: 10.15514/ISPRAS-2020-32(5)-12.
- [26] Лаврищева Е.М., Петренко А.К. и др. Технология сборочного создания экспериментального варианта ядра OS Linux с обеспечением качества для применения в прикладных и технических системах. Системный администратор, вып. 11, 2021 г., стр. 80-89 / Lavrischeva E.M., Petrenko A.K. et al. Technology of assembly creation of an experimental version of the OS Linux kernel with quality assurance for use in applied and technical systems. System Administrator, vol. 11, 2021, pp. 80-89 (in Russian).
- [27] Лаврищева Е.М., Петров И.Б. Технология моделирования физических и математических задач предметных областей знаний. Евразийское научное объединение, no. 1-1, 2021 г., стр. 35-43 / Lavrischeva E.M., Petrov I.B. Technology for modeling physical and mathematical problems of applied fields of knowledge. Eurasian Scientific Association, no. 1-1, 2021, pp. 35-43 (in Russian).
- [28] Lavrischeva E.M., Petrenko A.K., Kozin S.V. Technology of assembly creation of an experimental version OS Linux kernels with quality assurance for applied and subject areas of knowledge. In Scientific Articles Collection of the 81st International Scientific Conference of Eurasian Scientific Association, 2021, pp. 94-102.
- [29] Лаврищева Е.М., Петров И.Б., Петренко А.К. Парадигмы моделирования и программирования задач предметных областей знаний: монография. Директ-Медиа, 2021 г., 496 стр. / Lavrischeva E.M., Petrov I.B., Petrenko A.K. Paradigms of modeling and programming tasks of subject areas of knowledge: monograph. Direct-Media, 2021, 496 p. (in Russian)

Информация об авторах / Information about authors

Екатерина Михайловна ЛАВРИЩЕВА – доктор физико-математических наук, профессор, главный научный сотрудник ИСП РАН, профессор МФТИ, лауреат премии Совета Министров СССР (1985). Область интересов: надежность и качество, моделирование сложных систем, Web-системы, сборочное программирование.

Ekaterina Mikhailovna LAVRISCHEVA – Doctor of Physical and Mathematical Sciences, Professor, Principal Researcher at ISP RAS, Professor at MIPT, Laureate of the USSR Cabinet of Ministers Prize (1985). Areas of interest: reliability and quality, modeling of complex systems, Web-systems, assembly programming.

Александр Константинович ПЕТРЕНКО – доктор физико-математических наук, профессор, заведующий отдела ИСП РАН, профессор МГУ и ВШЭ. Область интересов: формальные методы программной инженерии, тестирование программного и аппаратного обеспечения, формальная спецификация требований.

Alexander Konstantinovich PETRENKO – Doctor of Physical and Mathematical Sciences, Professor, Head of the Department of ISP RAS, Professor of MSU and HSE. Areas of interest: formal methods of software engineering, testing of software and hardware, formal specification of requirements.