УДК 004.05

А.В. Баев, А.В. Самонов, В.М. Сафонов, С.В. Краснов, С.Р. Малышев

МЕТОДИКА ПРОЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ПРОИЗВОДСТВЕННЫМИ И ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ НА ОСНОВЕ UML-ПРОФИЛЯ ДЛЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ MARTE

Баев Алексей Владимирович, *окончил Военно-космическую академию (ВКА) им. А.Ф. Можайского. Научный сотрудник ВКА им. А.Ф. Можайского. Имеет более 10 трудов в области системного анализа, системной и программной инженерии, валидации и верификации программного обеспечения, методов и средств обеспечения информационной безопасности. [e-mail: baih@mail.ru].*

Самонов Александр Валерьянович, *кандидат технических наук, доцент, окончил Военную инженерную академию им. А.Ф. Можайского. Старший научный сотрудник ВКА 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].*

Сафонов Вадим Максимович, *кандидат технических наук, окончил ВКА им. А.Ф. Можайского. Начальник лаборатории ВКА им. А.Ф. Можайского. Имеет более 10 трудов в области системной и программной инженерии, моделирования и тестирования программного обеспечения. [e-mail: safonov-vm@mail.ru].*

Краснов Сергей Васильевич, *кандидат технических наук, доцент, окончил Ульяновское высшее военное командное училище связи, адъюнктуру Ульяновского высшего военного инженерного училища связи. Доцент Высшей школы бизнес-инжиниринга 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].*

Малышев Сергей Романович, *кандидат технических наук, доцент, окончил Военный инженерный Краснознаменный институт им. А.Ф. Можайского. Заслуженный изобретатель РФ, доцент Военной академии связи 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].*

Аннотация

В статье представлены методы и средства специфицирования, проектирования, верификации и валидации технических решений при разработке автоматизированных систем управления производственными и технологическими процессами (АСУ ТП) на ранних стадиях их создания – анализе требований и проектировании. Для разработки формализованного описания состава, структуры и алгоритмов функционирования аппаратных и программных компонентов АСУ ТП использованы языки визуального моделирования FUML, SysML, VSL, UML-профиль для встроенных систем реального времени MARTE и созданные на их основе специальные шаблоны проектирования. Проектные

решения, определяющие состав, структуру и алгоритмы функционирования программно-аппаратной части системы, представляются в виде комплекса структурных и поведенческих UML-диаграмм, аннотированных описанием эксплуатационно-технических характеристик входящих в ее состав компонентов. Представлены методы и средства разработки модели рабочей нагрузки, используемой для моделирования и анализа характеристик оперативности, производительности и масштабируемости АСУ ТП в различных режимах и условиях функционирования. Описана методика моделирования и анализа проектных решений посредством имитационного моделирования в среде CPN Tools с использованием математического аппарата сетей Петри.

Ключевые слова: автоматизированные системы управления технологическими процессами, имитационное моделирование, оперативность, производительность, сети Петри, UML-профиль MARTE.

doi: 10.35752/1991-2927_2023_3_73_54

METHODS OF DESIGNING THE AUTOMATED SYSTEMS OF MANAGEMENT FOR MANUFACTURING PROCESSES BASED ON THE UML PROFILE FOR MARTE REAL-TIME SYSTEMS

Aleksei Vladimirovich Baev, *graduated from Mozhaisky Military Space Academy; Staff Scientist of Mozhaisky Military Space Academy; an author of more than 10 works in the field of systems analysis, system and software engineering, validation and verification of software, methods and means of ensuring information security. e-mail: baih@mail.ru.*

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

Vadim Maksimovich Safonov, *Candidate of Sciences in Engineering; graduated from Mozhaisky Military Space Academy; Head of the laboratory in Mozhaisky Military Space Academy; an author of more than 10 works in the field of system and software engineering, modeling and software testing. e-mail: safonov-vm@mail.ru.*

Sergei Vasilevich Krasnov, *Candidate of Sciences in Engineering, Associate Professor; graduated from the Ulyanovsk Higher Military Command School of Communications; Adjunct of Ulyanovsk Higher Military Engineering Communications School; Associate Professor at Graduated School of Business Engineering 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.*

Sergei Romanovich Malyshev, *Candidate of Sciences in Engineering, Associate Professor; graduated from the Mozhaisky Military Institute of Engineering; Honored Inventor of the Russian Federation; Associate Professor at 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.*

Abstract

The article presents methods and means to specify, design, verify and validate technical solutions when developing the automated systems of management for manufacturing processes at the stages of analysing requirements and making the design. The visual modeling languages such as FUML, SysML, VSL, UML profile for embedded real-time systems MARTE and special design patterns created on their basis were used to develop a formalized description of the composition, structure and algorithms for the hardware and software functioning. Design solutions that determine the composition, structure and algorithms of the software and hardware functioning of the system are presented as a complex of structural and behavioral UML diagrams annotated with a description of the operational and technical characteristics of its components. Methods and tools for developing a workload model used for modeling and analyzing the characteristics of efficiency, performance and scalability of the system in various modes and operating conditions are presented. The methods of modeling and analysis of design solutions by means of simulation modeling in the CPN Tools environment using the mathematical apparatus of Petri nets is described.

Keywords: automated systems of management for manufacturing processes, simulation modeling, efficiency, performance, Petri net modeling, UML profile MARTE.

Введение

Автоматизированные системы управления производственными и технологическими процессами (АСУ ТП) представляют собой комплекс программных и программно-аппаратных средств, предназначенных для контроля за технологическим и (или) производственным оборудованием (исполнительными устройствами) и производимыми ими процессами, а также для управления такими оборудованием и процессами [1]. Основными характеристиками качества АСУ ТП являются: надежность, производительность, безопасность и защищенность. Надежность – свойство системы сохранять во времени в установленных пределах значения всех параметров, характеризующих ее способность выполнять требуемые функции в заданных режимах и условиях применения [2]. Для обеспечения требуемого уровня надежности система должна обладать соответствующей производительностью и масштабируемостью. Вопросы, связанные с обеспечением требуемого уровня качества функционирования АСУ ТП, необходимо решать комплексно и системно на всех стадиях их жизненного цикла: обоснования требований, проектирования, разработки, испытаний и эксплуатации. При этом особое внимание следует уделить этапам обоснования требований и проектирования, на которых создаются фундаментальные основы будущей системы.

На этих этапах необходимо корректно определить требования к функциональным возможностям и эксплуатационно-техническим характеристикам АСУ ТП, спроектировать и разработать оптимальный состав и структуру входящих в нее аппаратных и программных компонентов, проанализировать и оценить соответствие проектных решений заданным требованиям и условиям применения системы. Это позволит снизить риски создания системы, не отвечающей заданным требованиям, а также существенно повысить качество управления процессом ее жизненного цикла. Для этого необходимо создать единую модельно-языковую и информационно-программную среду и средства автоматизации соответствующих процессов. Примерами реализации данного подхода являются современные технологии промышленной разработки сложных программно-технических систем Rational Unified Process (RUP) [3], Microsoft Solution Framework (MSF) [4], SADT (IDEFx) [5].

Целью исследований и работ, результаты которых представлены в данной статье, является развитие и совершенствование модельно-ориентированных технологий и методов, разработка методического, модельного, алгоритмического и программного обеспечения процессов проектирования АСУ ТП. В данной статье представлены методы и средства формализованного описания состава, структуры и алгоритмов функционирования аппаратных и программных компонентов АСУ ТП с помощью языков визуального моделирования UML, SysML, VSL и UML-профиля систем реального времени MARTE (Modeling and Analysis of Real-Time Embedded Systems) [6].

Материал статьи организован следующим образом. В первом разделе представлены краткая характеристика, назначение, структура и способы применения профиля MARTE для разработки сложных аппаратно-программных систем. Во втором разделе описаны методы и средства проектирования АСУ ТП с помощью моделей и средств данного профиля. В третьем разделе представлена методика разработки модели рабочей нагрузки, используемой для моделирования и анализа характеристик оперативности и производительности АСУ ТП. В четвертом разделе представлена методика моделирования и анализа проектных решений посредством имитационного моделирования в среде CPN Tools с использованием математического аппарата сетей Петри. В заключении подведены итоги текущих исследований, определены направления развития и совершенствования изложенного в статье подхода, методов и средств.

1 Назначение, структура и способы применения профиля MARTE

UML-профиль MARTE предназначен для проектирования сложных аппаратно-программных систем, к которым предъявляются высокие требования к оперативности, производительности и надежности функционирования. Спецификация MARTE представлена в виде четырех пакетов (рис. 1) [6].

Первый пакет MARTE Foundations содержит базовые типы данных для описания количественных и качественных характеристик систем – CoreElements и NFP (Non-Functional Properties), времени и временных отношений – Time, программных и аппаратных ресурсов – GRM (GeneralResourceModeling), а также их пространственного и временного размещения – Alloc (Allocation Modeling). Второй пакет MARTE Design Model включает средства для высокоуровневого и детального проектирования систем: GSM (Generic Component Mode), HLAM (High-Level Application Modeling), SRM (Software Resource Modeling) и HRM (Hardware Resource Modeling). Третий пакет MARTE Analyze Model содержит модели и средства для анализа проектных решений. Функции пакета MARTE Analyze Model разбиты на базовую общую часть – пакет GQAM (Generic Quantitative Analysis Modeling) и два пакета для конкретных областей анализа: SAM (Schedulability Analysis Modeling) – анализа механизмов планирования и PAM (Performance Analysis Modeling) – анализа производительности (использование вычислительных мощностей, оперативной и внешней памяти, энергопотребления и др.). Четвертый пакет MARTE annexes Model включает описание предметно-ориентированного языка VSL (Value Specification Language), библиотеку функций MARTE Library для построения, моделирования и оценивания эксплуатационно-технических характеристик (ЭТХ) разрабатываемых систем, а также профиль для проектирования отказоустойчивыхсистем со структурной избыточностью – RSM (Repetitive Structure Modeling). Профиль MARTE определяет точную семантику для моделирования событий, времени

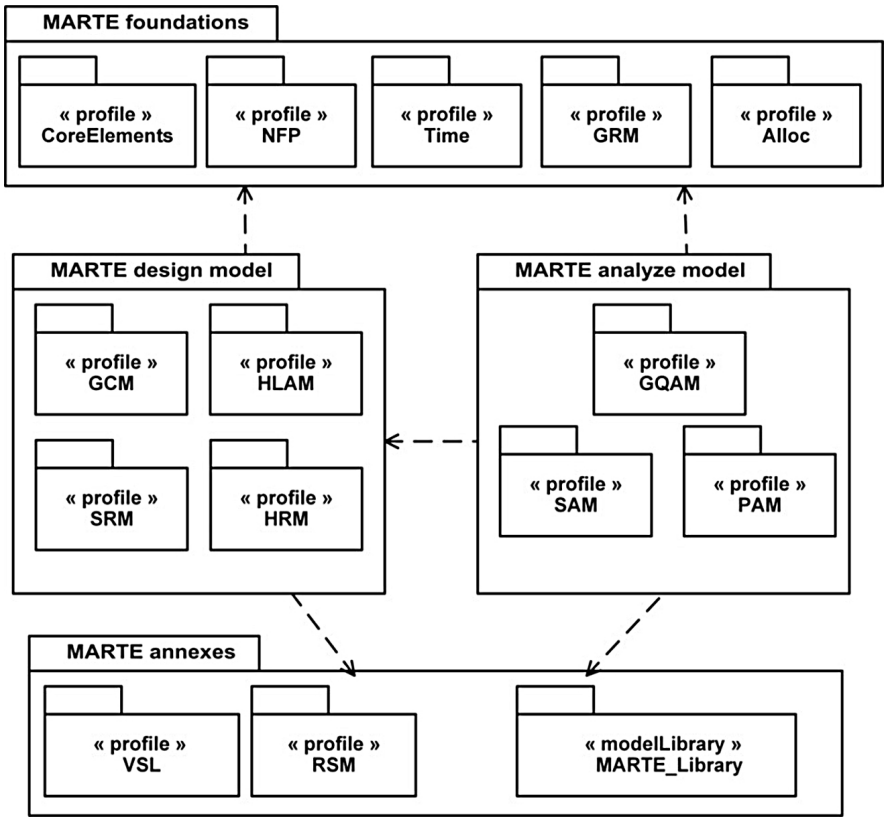

Рис. 1. Состав и структура профиля MARTE

и ресурсов. Это позволяет автоматически преобразовывать модели, созданные на основе спецификации MARTE, в модели более низкого уровня абстракции, такие как UML или AADL для систем на кристалле (SoC), или в программный код на языках C и C++.

В самом обобщенном виде методика проектирования и анализа характеристик качества АСУ ТП на основе UML-профиля MARTE включает:

1) разработку проекта системы (определение состава, структуры и алгоритмов функционирования программных и аппаратных средств) с помощью средств, представленных в пакетах профиля MARTE: Foundation, GCM, HLAM, SRM, HRM, VSL, MARTE_Library;

2) генерацию рабочих нагрузок, соответствующих штатным и нештатным режимам функционирования АСУ ТП, с помощью средств, представленных в пакетах: GRM, SRM, HRM, GOAM, SAM, PAM, VSL, MARTE_Library;

3) моделирование и анализ проектных решений АСУ ТП с целью определения рационального состава, структуры и алгоритмов функционирования.

Обобщенная схема реализации данной методики представлена на рисунке 2.

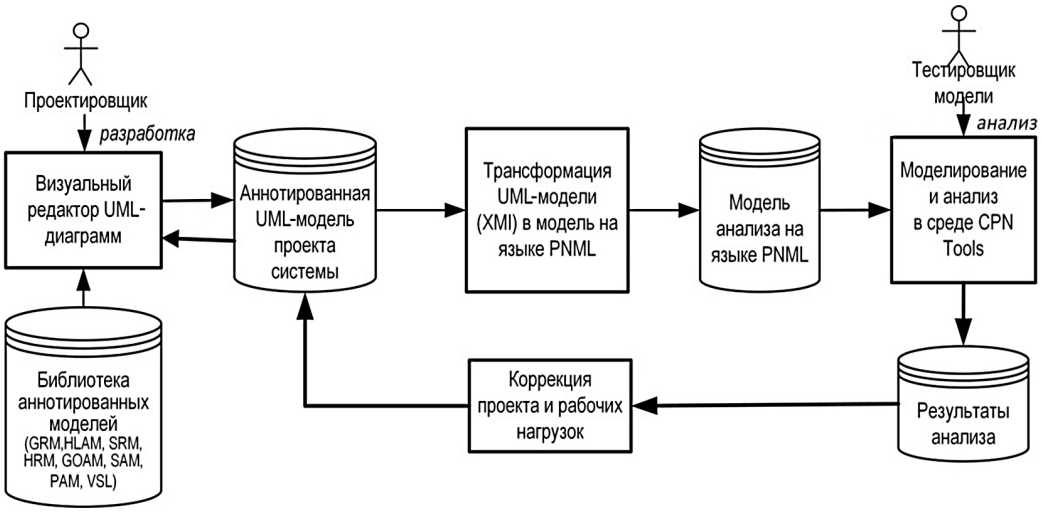

Рис. 2. Обобщенная схема реализации методики проектирования АСУ ТП

2 Методы и средства разработки проекта АСУ ТП

Методика разработки проектных решений АСУ ТП включает следующие этапы [7]:

- построение модели комплекса требований АСУ ТП в виде совокупности диаграмм вариантов использования (d_uc) и диаграмм классов (d_class);
- описание алгоритмов реализации функциональных возможностей АСУ ТП с помощью специальных шаблонов, построенных на основе диаграмм поведения (d_act, d_seq, d_sm);
- разработку системной архитектуры АСУ ТП в виде совокупности диаграмм размещения (d_alloc), блоков (d_block), диаграмм классов (d_class);
- описание требований к ЭТХ АСУ ТП на языке объектных ограничений OCL (Object Constraint Language);
- верификацию и валидацию разработанных архитектуры и проектных решений АСУ ТП.

Наиболее важными концепциями при разработке проектных решений АСУ ТП на основе UML-профиля MARTE являются: события, действия, ресурсы и время. События вызывают выполнение определенных действий, которые используют соответствующие ресурсы. Ресурсы могут быть программными или аппаратными. События и действия связаны с моментами и интервалами времени. Функциональные элементы АСУ ТП должны быть распределены в пространстве и во времени по имеющимся ресурсам. Пространственное распределение означает сопоставление сервисов (функций) аппаратным и программным средствам системы. Временное распределение означает временную упорядоченность вычислений и использования ресурсов. Системная архитектура АСУ ТП представляет собой описание состава, структуры и характеристик комплекса программных и аппаратных средств, а также их связей и способов взаимодействия и функционирования.

АСУ ТП может включать в свой состав следующие типы аппаратных средств: компьютеры, коммутационное и сетевое оборудование, хранилища данных, средства защиты, принтеры, системы питания и др. В терминологии MARTE эти средства называются ресурсами. Для описания аппаратных средств (ресурсов) разработаны соответствующие шаблоны в виде диаграмм классов. На рисунке 3, в частности, представлена диаграмма классов, описывающая один из основных компонентов АСУ ТП – вычислительный ресурс (ComputingResource).

ComputingResource может быть реализован в виде универсальной ЭВМ (HW_Processor), специализированного аппаратного комплекса (HW_ASIC), настраиваемого (HW_PLD) и др. В таблице 1 представлены атрибуты класса HwProcessor, предназначенного для описания его центрального компонента – процессорного блока.

Важное значение при проектировании АСУ ТП имеет разработка алгоритмов и средств планирования и управления вычислительным процессом. Для описания и моделирования процессов планирования разработана представленная на рисунке 4 диаграмма классов. Основными элементами данной диаграммы являются следующие классы: Scheduler (Планировщик), ProcessingResource (планируемые ресурсы), SchedulingPolicy (политика планирования), SchedulableResource (совместно используемые ресурсы), MutualExclusionResource (ресурсы, управляемые протоколом взаимного исключения).

Класс ProcessingResource обобщает понятия ComputingMedia, ComputingResource и DeviceResource. Атрибут speedFactor определяет, какая часть от общего объема ресурса используется. SchedulingPolicy и SchedulableResource используются для описания механизмов планирования ресурсов при одновременном обращении к ним параллельно выполняющимися процессами. Базовые механизмы планирования реализуются средствами синхронизации (семафор, мьютекс, события, таймеры и др.) в соответствии с определенным протоколом защиты. В случае иерархического планирования планировщики, отличные от основного планировщика, могут быть описаны классом SecondaryScheduler.

Таблица 1

Основные атрибуты класса HwProcessor

Атрибут класса HwProcessor	Описание атрибута
architecture: NFP_DataSize	архитектура процессора
caches: HwCache[0..*]	уровни и размеры кэша процессора
ownedMMUs: HwMMU[0..*]	блоки управления памятью
ownedISAs: HwISA[1..*]	архитектура набора команд
mips: NFP_Natural	пропускная способность процессора
ipc: NFP_Real	количество инструкций, выполняемых за каждый такт
nbCores: NFP_Natural	количество ядер
nbPipelines: NFP_Natural	количество нитей на ядро
nbStages: NFP_Natural	количество этапов в конвейере
nbALUs: NFP_Natural	количество арифметико-логических блоков
nbFPUs: NFP_Natural	количество блоков с плавающей запятой

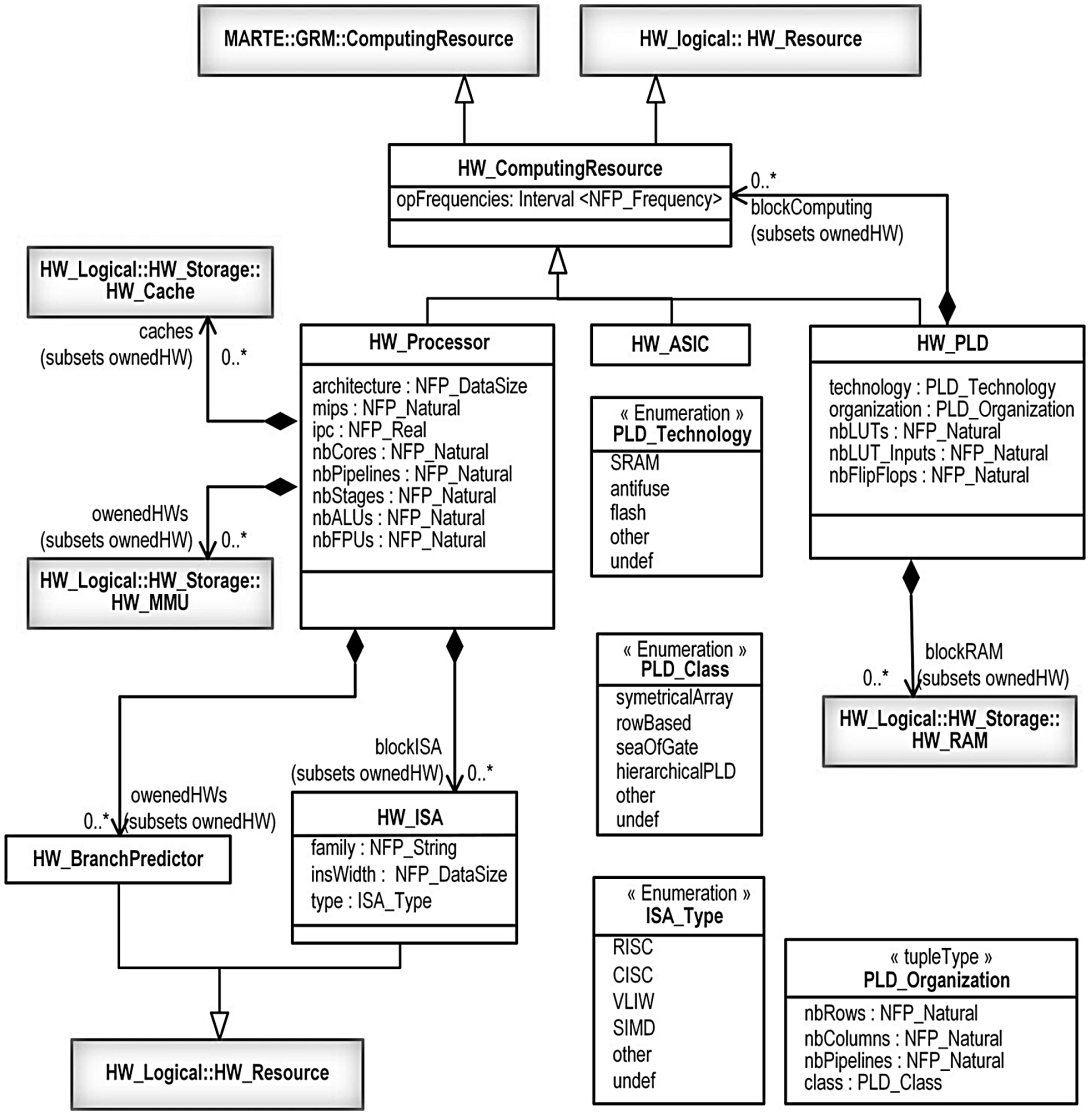

Рис. 3. Диаграмма классов, описывающая ComputingResource

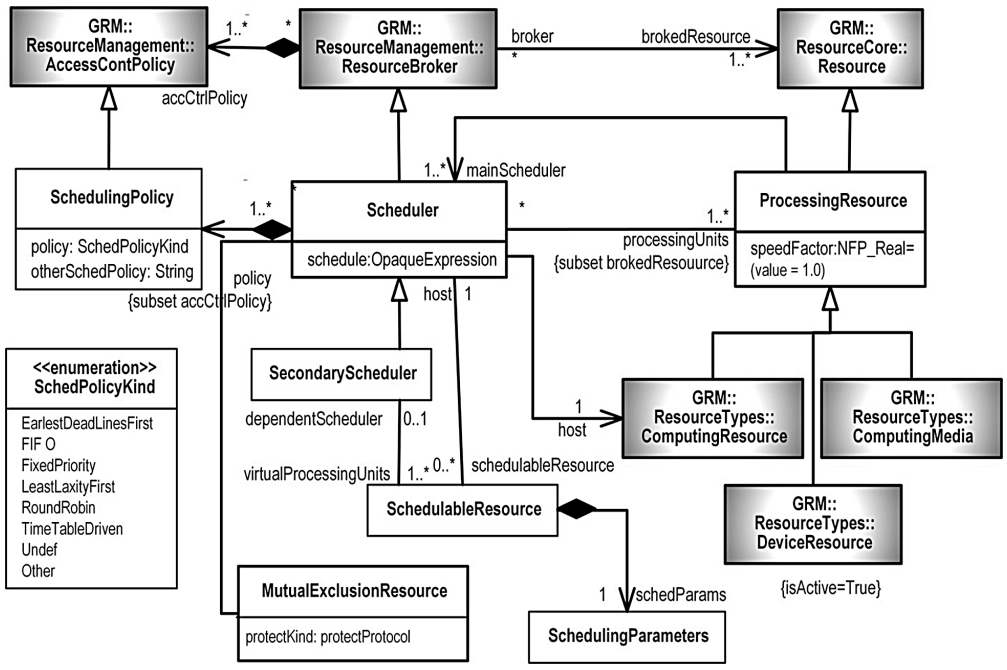

Рис. 4. Диаграмма классов, используемая для описания и моделирования процессов планирования

Для идентификации общего ресурса, механизма его защиты и связанного с ним протокола используется класс MutualExclusionResource, реализующий механизм взаимного исключения.

Таким образом, в ходе проектирования определяются состав, структура и алгоритмы функционирования программно-аппаратной части системы в виде комплекса структурных (d_class, d_block, d_alloc) и поведенческих диаграмм (d_act, d_seq, d_sm), аннотированных описанием ЭТХ входящих в ее состав компонентов. Для верификации и валидации проектных решений посредством имитационного моделирования необходимо разработать рабочие нагрузки. Описание методов и средств решения данной задачи представлено в следующем разделе.

3 Разработка модели рабочей нагрузки для моделирования и анализа характеристик оперативности и производительности АСУ ТП

Схема, иллюстрирующая состав и структуру моделей и средств формирования и реализации рабочей нагрузки в ходе моделирования и анализа проектных решений АСУ ТП, представлена на рисунке 5.

Как видно из рисунка 5, рабочая нагрузка WorkloadBehavior описывается с помощью класса BehaviorScenario (сценарий), выполнение которого инициируется потоком событий, представленных классом WorkloadEvent. Поток событий может быть сгенерирован и представлен следующими способами:

- потоком событий, хранящимся в файле (*EventTrace*);
- в соответствии с определенным вероятностным законом (*ProbDistribution*);
- с помощью генератора рабочей нагрузки, который может быть смоделирован как конечный автомат (*WorkloadGenerator*);
- в виде событий, возникающих в соответствии с определенным временным графиком (*TimeEvent*).

При реализации сценария (*BehaviorScenario*) используются аппаратные и программные ресурсы. К аппаратным относятся: процессоры и ядра, оперативная память, внешние носители данных, аппаратура и каналы передачи данных, источники питания и др. К программным ресурсам относятся: процессы, критические секции, семафоры, блокировки, пулы буферов и др. Тип ресурсов, объемы и время их использования при реализации сценария описываются атрибутами класса BehaviorScenario, представленными в таблице 2.

Для описания требований к составу и характеристикам используемых ресурсов также применяются атрибуты класса ResourceUsage, потомком которого является класс BehaviorScenario (табл. 3).

Таблица 2

Атрибуты класса BehaviorScenario

Атрибут	Описание
hostDemand: NFP_Duration	Время использования хоста, требуемое для реализации сценария
hostDemandOps: NFP_Real[*]	Использование хоста, выраженное в операциях
interOccTime	Временной интервал между двумя последовательными событиями
throughput [*]	Пропускная способность
respTime	Задержка, включая первоначальную задержку планирования
utilization	Среднее число экземпляров сценария, активных в любой момент времени
utilizationOnHost	Доля времени, в течение которого хост занят выполнением сценария
root Step	Первый шаг сценария
timing: TimingObs[*]	Временные наблюдатели, связанные с этим сценарием

Таблица 3

Атрибуты класса ResourceUsage

Атрибут	Описание
execTime: NFP_Duration {ordered} [*]	Задержка без учета начальной задержки планирования
msgSize: NFP_DataSize {ordered} [*]	Объем данных, передаваемых ресурсом
allocatedMemory: NFP_DataSize {ordered} [*]	Объем памяти, который запрашивается у ресурса или возвращается ему
usedMemory: NFP_DataSize {ordered} [*]	Объем памяти, который будет использоваться
powerPeak:NFP_Power {ordered} [*]	Мощность, которая должна быть доступна
energy:NFP_Energy {ordered} [*]	Количество энергии, которое будет потребляться

Сценарий (*BehaviorScenario*) представляет собой определенную последовательность шагов (Step), каждый из которых также может быть сценарием. Для реализации каждого шага используется определенный вычислитель (*ExecutionHost*) и процесс операционной системы (*SchedulableResource*). На шаге сценария могут выполняться следующие операции: получение (*AcquireStep*) и освобождение (*ReleaseStep*) ресурса, передача данных (*CommunicationStep*) и запрос сервиса (*RequestedService*). Последовательность выполнения шагов сценария описывается с помощью отношения *PrecendenceRelation*, определяющего тип связи между шагами (*ConnectKind*). Как видно из рисунка 5, возможны следующие типы связей:

- *sequence* – простая последовательность;
- *branch* – ветвь, состоящая из нескольких шагов, каждый из которых может осуществляться с определенной вероятностью;

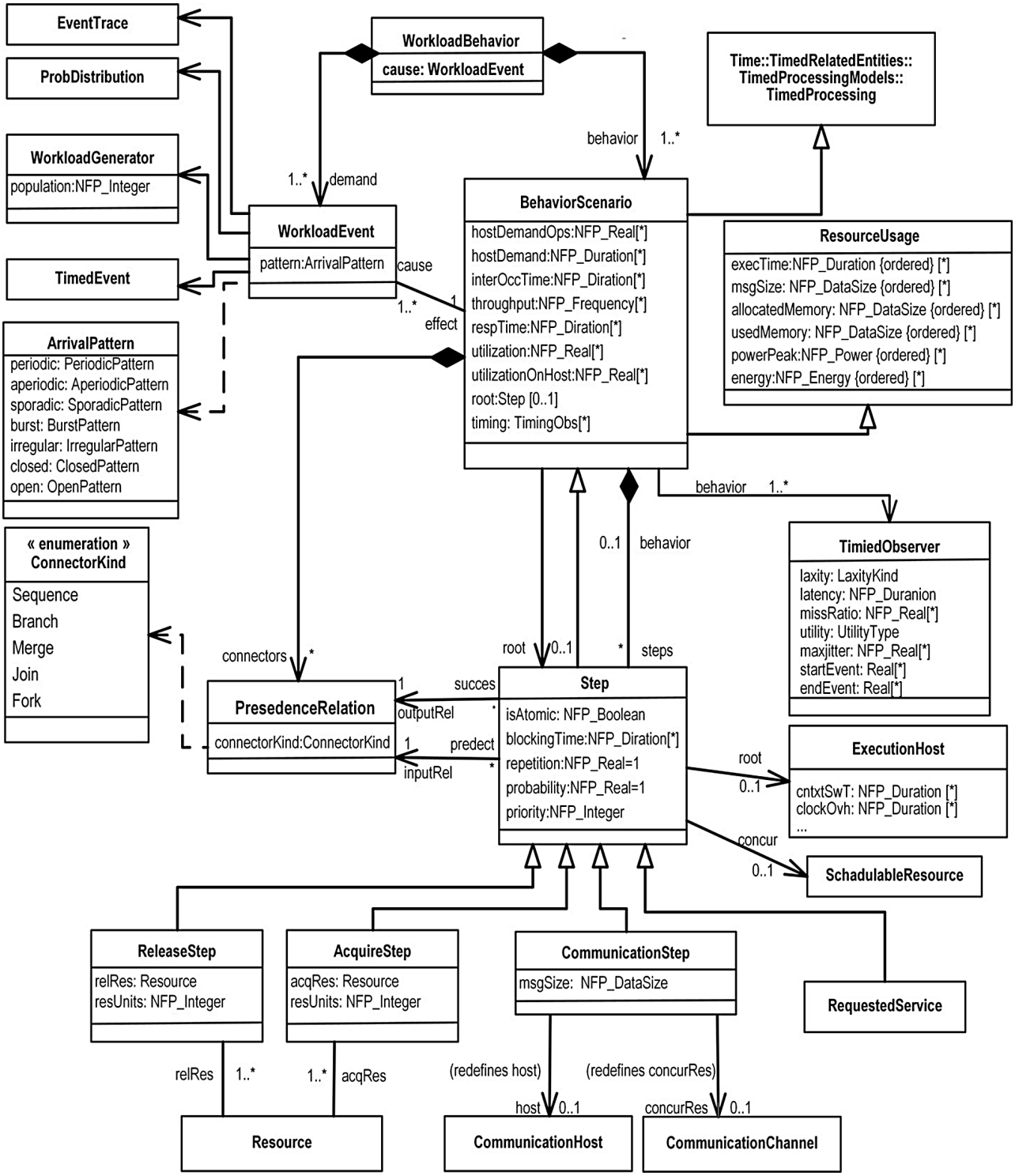

Рис. 5. Состав и структура моделей и средств формирования и реализации рабочей нагрузки

- *merge* – слияние нескольких шагов в один, инициированный любым из предшествующих;

- *fork* – один шаг создает несколько последующих шагов, которые выполняются параллельно;

- *join* – объединение нескольких предшествующих шагов в один, запускаемый после завершения всех предшественников.

Шаг может быть необязательным и выполняться с некоторой вероятностью или повторяющимся определенное количество раз. Свойство «isAtomic» определяет атомарность выполнения (по умолчанию равно false). Шаг имеет ассоциацию с хостом *hostDemand* (*ExecutionHost*) и планируемым ресурсом (*SchedulableResource*).

CommunicationStep описывает передачу сообщений между объектами и имеет атрибут, определяющий размер сообщения *msgSize*. Класс «*ExecutionHost*» описывает используемый вычислительный ресурс посредством атрибутов, представленных в таблице 4.

TimedObservers описывает средства, с помощью которых определяются требования и рассчитываются прогнозные значения ЭТХ АСУ ТП на интервале времени между событиями *startEvent* и *endEvent*. Перечень атрибутов TimedObserver может быть расширен, чтобы определить данные (атрибуты, параметры), которые он должен собирать в процессе моделирования и наблюдения. Например, можно включить атрибуты для описания энергопотребления (NFP_Energy), использования памяти (HwMemory) и др.

Разработанные с помощью представленных в данном разделе моделей и средств варианты рабочих нагрузок используются для анализа проектных решений и их доработки в случае необходимости. Методика и пример решения данной задачи с помощью средств CPN Tools представлены в следующем разделе.

4 Методика моделирования и анализа проектных решений АСУ ТП

В настоящее время для анализа и оценивания характеристик производительности и масштабируемости программно-технических систем используются сети массового обслуживания (CeMO) [8], стохастические алгебры процессов [9], сети Петри [10–14], имитационные модели [15, 16]. Как показал проведенный анализ представленных выше источников, наиболее перспективным представляется использование математического аппарата сетей Петри и средств, использующих эти модели.

Моделирование и анализ проектных решений осуществляется в среде специальной моделирующей системы, которая использует математический аппарат сетей Петри – CPN Tools. В CPN Tools реализованы средства для описания и моделирования формальных моделей, называемых иерархическими временными раскрашенными сетями Петри (ИВРСП). Раскрашенная сеть Петри на множестве типов позиций U представляет собой кортеж $(P,T,F,\upsilon,\rho,\Lambda)$,

где P и T – непересекающиеся конечные множества позиций и переходов;

$F \subseteq (P \times T)\left[(T \times \cup P)\right.$ – множество дуг;

$\upsilon : P \to 2^U$ – функция типов позиций, отображающая элементы P в подмножества U;

$\rho : F \to Expr$ – функция меток дуг;

$\Lambda : T \to Lab \cup \{\tau\}$ – функция меток переходов.

Для раскрашенной сети $N = (P,T,F,\upsilon,\rho,\Lambda)$ на множестве U разметка N есть функция $m : P \to N^U$, такая, что $m(p) \in \mathbb{N}^{\upsilon(p)}$ для $\rho \in P$. Пара $(N; m)$ называется размеченной сетью Петри. Переход $t \in T$ активен в разметке m тогда и только тогда, когда $\exists \forall p \in P : (\rho,t) \in F \Rightarrow m(\rho) \geq \mathcal{T}(p(p,t),\upsilon)$.

Активный переход t может сработать, порождая новую разметку $m'(p) = m(p) - -\mathcal{T}(\rho(p,t),\upsilon) + \mathcal{T}(\rho(t,p),\upsilon)$ для каждого $\rho \in P$ (обозначается $m \xrightarrow{t} m'$). Размеченная раскрашенная сеть определяет систему переходов, представляющую наблюдаемое поведение моделируемой с ее помощью системы.

ИВРСП называются раскрашенными, так как токены (маркеры) могут иметь тип, для графического представления которого используются разные цвета. Иерархические сети обеспечивают построение сложных моделей,

Таблица 4

Атрибуты класса «ExecutionHost»

Атрибут класса ExecutionHost	Описание атрибута
commTxOvh: NFP_Duration [*]	Интервал времени хоста на отправку сообщений
commRcvOvh:NFP_Duration [*]	Интервал времени хоста на прием сообщений
cntxtSwT: NFP_Duration [*]	Время переключения контекста
clockOvh: NFP_Duration [*]	Накладные расходы
schedPriRange: NFP_Interval [*]	Диапазон приоритетов, предлагаемых этим процессором
memSize: NFP_DataSize [0..1]	Размер памяти
utilization: NFP_Real [*]	Загрузка процессора
throughput: NFP_Frequency [*]	Пропускная способность хоста

в которых любой элемент может быть представлен сетью Петри. Временные сети Петри используют понятие модельного времени для описания продолжительности действий в реальных системах. В отличие от классических сетей Петри, где срабатывание перехода происходит мгновенно, срабатывание перехода во временной сети связано с определенной продолжительностью или временной задержкой. Это позволяет анализировать временные характеристики реальных объектов: время отклика, задержку в обслуживании запросов и т. п. ИВРСП эквивалентны машине Тьюринга и представляют собой универсальную алгоритмическую систему [10].

Анализ проектных решений АСУ ТП осуществляется посредством имитационного моделирования с помощью средств, предоставляемых инструментом CPN Tools. Для реализации данного подхода необходимо [15]:

- определить цели, методы, критерии и порядок проведения и использования результатов моделирования;

- преобразовать представленные в виде комплекса структурных и поведенческих диаграмм проектные решения АСУ ТП в имитационную модель системы на языке PNML;

- подготовить рабочую нагрузку и другие исходные данные для проведения имитационного моделирования;

- разработать и отладить в среде CPN Tools средства реализации процессов моделирования, сбора и анализа данных;

- организовать и провести модельные эксперименты;

- выполнить анализ полученных результатов, сделать соответствующие выводы и разработать предложения по доработке проектных решений в случае необходимости.

Основными целями имитационного моделирования при анализе проекта АСУ ТП являются: оценка ожидаемой производительности системы и определение таких параметров системы, при которых она будет функционировать оптимальным образом.

Для преобразования представленного в виде комплекса UML-диаграмм проекта АСУ ТП в имитационную модель инструмента CPN Tools разработана диаграмма классов, описывающая метамодель сети Петри (рис. 6).

Корневой узел pnml представляет PNML-документ сети Петри, расширенный классом net. Основными элементами сети Петри являются place (место), transition (переход) и arc (дуга), представленные в метамодели PNML соответствующими классами. Положение каждого объекта в документе pnml управляется графическим классом, который, в свою очередь, состоит из двух классов offset и position для представления относительного и фактического положения каждого объекта. Класс

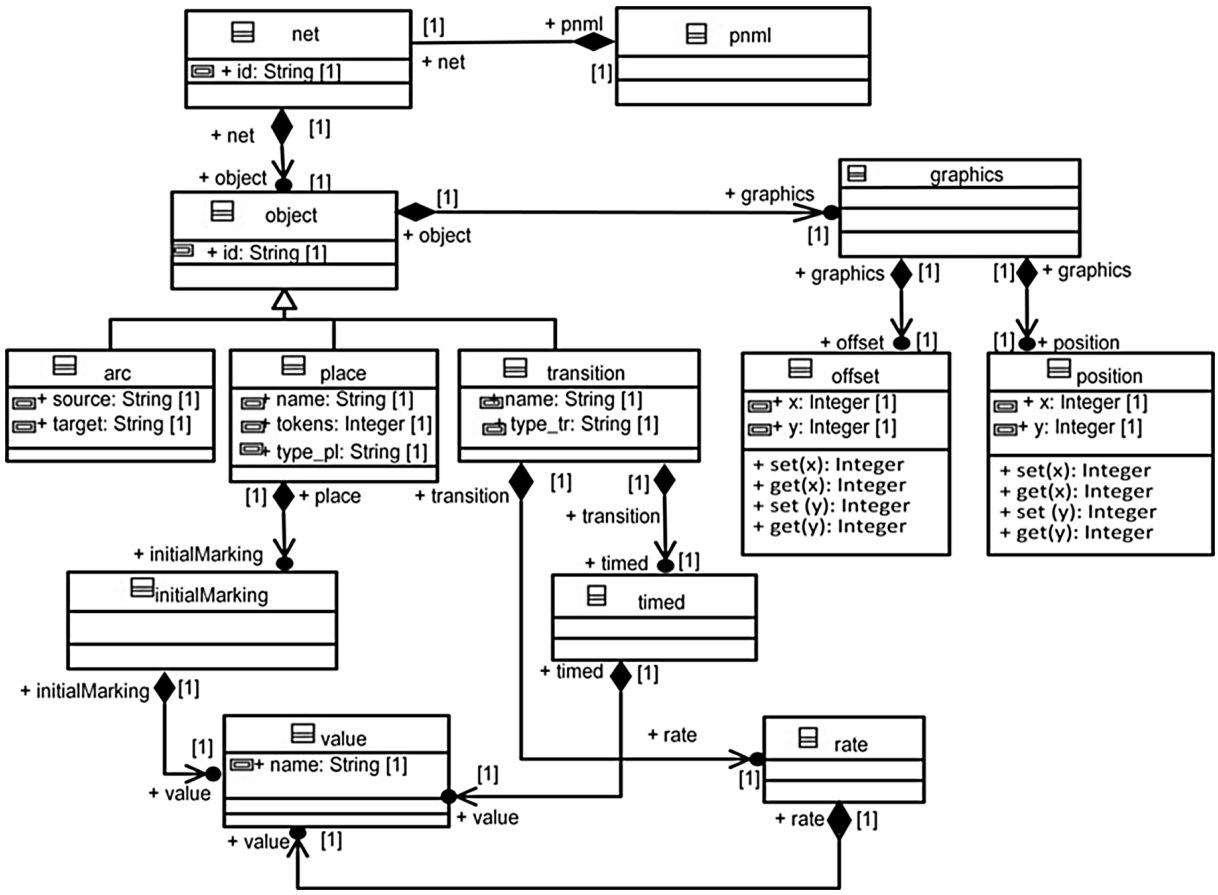

Рис. 6. Метамодель сети Петри на языке PNML

метамодели PNML initialMarking представляет информацию о маркерах сети Петри, а класс timed используется для представления двух типов переходов timed и immediate. Класс value используется для представления количества токенов в позиции, временных интервалов и значений других атрибутов. Класс rate используется для представления скорости переходов.

Процессы непосредственного моделирования и анализа проектных решений АСУ ТП, реализуемые с целью оценивания ожидаемых характеристик оперативности и производительности, рассмотрим на примере системы, обобщенный алгоритм функционирования которой представлен на рисунке 7. Моделирование процессов функционирования системы происходит следующим образом. На вход системы подаются сгенерированные рабочие нагрузки (поток заданий), соответствующие как штатным, так и нештатным режимам работы. Варьируемыми параметрами рабочей нагрузки являются: время, необходимое для выполнения задания определенного типа, интервалы времени между заданиями, среднее количество заданий в единицу времени, потребности заданий в ресурсах: количестве вычислительных мощностей (процессоров, ядер, потоков), объемов и характеристик оперативной и внешней памяти, пропускной способности локальной сети.

Выполнение задания начинается тогда, когда ему будут доступны необходимые ресурсы: ЦП (центральный процессор), ОП (оперативная память), ВП (внешняя память), ДП (дисковая подсистема), СПД (система передачи данных). Объемы необходимых ресурсов определяются на основе анализа диаграмм классов и диаграмм поведения, образующих проект АСУ ТП, операционной среды, аппаратной платформы и других условий функционирования. Управление вычислительным процессом и ресурсами осуществляет планировщик (Sheduler). После завершения выполнения задания выделенные ресурсы освобождаются и передаются планировщиком (Sheduler) менеджеру ресурсов (Resource Manager).

Для сбора и обработки статистической информации о ходе и результатах выполнения потока заданий разрабатываются специальные утилиты (функции), называемые мониторами сбора данных (МСД). С помощью МСД собираются следующие параметры: длительность выполнения заданий, средняя длина очереди заданий, задержки при выполнении операций, время отклика, размеры передаваемых данных, частота поступления заданий, загрузка вычислителей, объемы используемой памяти, время и размеры использования других ресурсов. На основе этой информации могут быть рассчитаны комплексные показатели качества анализируемой системы, такие как вероятность достижения заданных значений характеристик оперативности, производительности и масштабируемости, а также эффективность использования программных и аппаратных ресурсов.

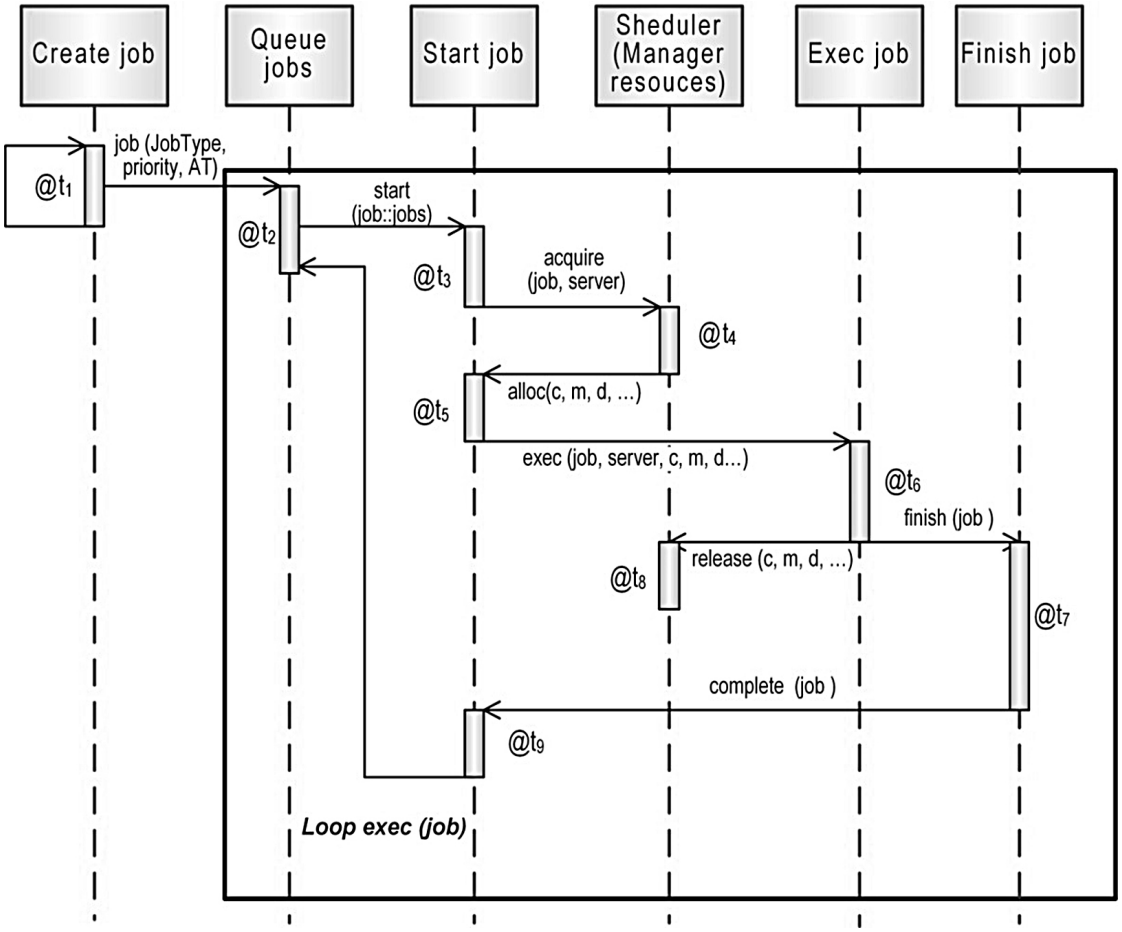

Рис. 7. Схема алгоритма функционирования системы в виде диаграммы последовательности

На рисунке 8 представлены разработанная в среде CPN Tools модель анализируемой системы и средства, используемые для моделирования и оценивания характеристик ее производительности и оперативности.

Модель состоит из двух модулей, которые представлены двумя переходами (Arrivals и Server) и двумя позициями (Queue и Completed). В начальной разметке список заданий пуст (1`[], см. рис. 8). Переходы Arrivals и Server детализируются в модулях второго уровня иерархии. Задания создаются и добавляются в очередь Queue в модуле Arrivals, а обрабатываются в модуле Server. Описание сети представлено в двух блоках объявлений (Declaration). Первый блок объявлений, называемый SYSTEM_DECLS, содержит описание типов данных, переменных и функций, используемых в анализируемой сети Петри. Второй блок, названный MONITOR_DECLS, содержит объявления, которые используются для описания мониторов.

Задания в зависимости от типа алгоритма, формата обрабатываемых данных, ресурсов, необходимых

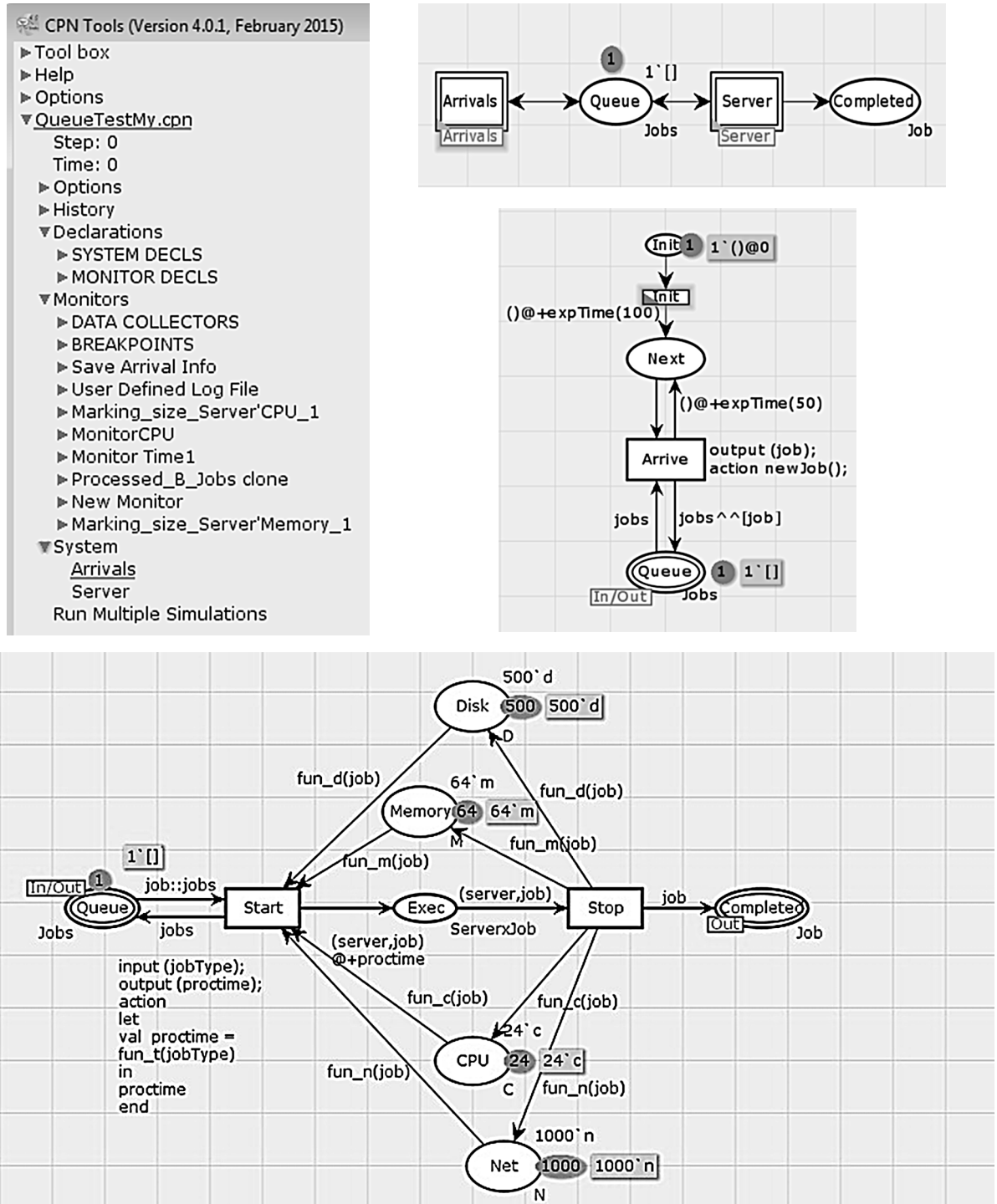

Рис. 8. Модель системы в среде CPN Tools

для их выполнения, и других характеристик делятся на типы. Для каждого типа задания формируется соответствующий набор атрибутов. Задание представляет собой запись: job = (jobType, priority, AT), где jobType – тип задания, priority – приоритет, AT – время создания. В CPN Tools имеются функции для моделирования следующих законов распределения случайной величины: экспоненциального, биномиального, нормального, бернули, гамма, пуассона, стьюдента и др. В нашем примере используется функция expTime, генерирующая временные задержки, распределенные по экспоненциальному закону с параметром mean. Функция newJob возвращает новое задание и время его запуска, устанавливаемое с помощью функции intTime. Тип заданий генерируется случайным образом с помощью функции run.

Для разработки рабочих нагрузок используется модель BehaviorScenario, представленная на рисунке 5 и в таблицах 2 и 3. Для реализации процессов моделирования и анализа в среде CPN Tools разработаны специальные функции определения, мониторинга и обработки данных. Описание этих функций представлено в таблице 5.

В модуле Server имеются два перехода (Start, Stop) и пять позиций (Exec, CPU, Memory, Disk, Net). Exec – соответствует состоянию сети Петри, когда задание выполняется. Позиции CPU, Memory, Disk и Net описывают состояние используемых при выполнении заданий ресурсов центрального процессора, оперативной памяти, дисковой памяти и системы передачи данных соответственно. Маркеры в этих позициях соответствуют объему доступных ресурсов. Варьируя этими значениями можно моделировать применение различных конфигураций аппаратной части системы обработки заданий.

Для того чтобы произошел переход Start, в очереди заданий должно быть хотя бы одно задание (job::jobs на дуге из позиции Queue), а также наличие соответствующего количества доступных ресурсов в позициях CPU, Memory, Disk и Net. Параметр @proctime определяет время выполнения задания определенного типа. ServerxJob (product Server * Job timed) используется для представления сервера, когда он занят обработкой задания job. Переход Stop станет разрешенным, когда сервер завершит обработку задания, т. е. когда отметка времени маркера в позиции Exec будет равна текущему модельному времени. Когда произойдет переход, выполненное задание будет добавлено в позицию Completed, а выделенные ему ресурсы освобождаются и возвращаются в соответствующие позиции CPU, Memory, Disk и Net.

Результаты имитационного моделирования сохраняются в журналах сбора данных о ходе и результатах моделирования. На основе этих данных формируются отчеты с результатами анализа производительности. Представленный в таблице 6 небольшой фрагмент из такого отчета содержит следующую информацию:

Таблица 5

Функции мониторов сбора данных

Идентификатор	Атрибут	Функция мониторов CPN Tools
t_job	hostDemand	fun_t(typeJob)
t_ob_op	hostDemandOps	fun_t_ops(typeJob)
t_int_event	interOccTime	expTime(mean)
t_util	utilization	dt/jobs
mem_alloc	allocatedMemory	fun_t_ops(typeJob)
net_alloc	allocatedNet	fun_t_ops(typeJob)
d_alloc	allocatedDisk	fun_t_ops(typeJob)

Таблица 6

Фрагмент отчета с результатами анализа производительности

Монитор	**Число измерений**	**Характеристика**	**Значение**	**0,5 ДДИ 95 %**	**0,5 ДДИ 99 %**
Monitor_Time	90	время выполнения задания	199,771689	17,447991	23,150121
Queue_Delay	95	задержка выполнения задания	101,836588	16,893574	22,414516
Queue_Length	95	длина очереди	4,443254	0,475698	0,629629
Marking_size_Server’CPU	100	загрузка ЦП	4,090476	0,066561	0,088099
Marking_size_Server’Memory	100	загрузка ОП	0,180952	0,133122	0,176199
Marking_size_Server’Disk	96	загрузка ДП	304,523810	3,328052	4,404972
Marking_size_Server’Net	80	загрузка СПД	820,400794	4,200127	5,559241
Server_Utilization	100	загрузка сервера	0,974603	0,022243	0,029441

название монитора, число измерений, среднее значение полученной с его помощью характеристики, половинные значения длин (ДДИ) 95-и 99-процентных доверительных интервалов соответствующих величин.

Как видно из этого отчета, средствами мониторов Monitor_Time, Queue_Delay и Queue_Length зафиксированы недопустимо высокие значения следующих характеристик: времени исполнения заданий, задержки выполнения заданий, длины очереди заданий. Анализ информации о загруженности ресурсов сервера, полученных с помощью мониторов разметки сети Петри: Marking_size_Server'CPU, Marking_size_Server'Memory, Server_Utilization, Marking_size_Server'Disk, Marking_size_Server'Net, позволяет сделать вывод о том, что причиной этого является недостаточный объем свободной ОП. Значение характеристики «загрузка ОП», равное 0,180952 Гб, соответствует среднему значению размера свободной памяти. Увеличение объема ОП с 8 до 12 Гб привело к получению допустимых значений задержки выполнения заданий до 4,9 с, длине очереди 0,27, при наличии свободной ОП в размере 2,5 Гб.

Анализ информации, получаемой в ходе имитационного моделирования, позволяет делать вывод о соответствии проектных решений установленным требованиям и условиям эксплуатации систем, определять, каких именно ресурсов недостаточно в данной конфигурации, а какие присутствуют в избытке. Реализация такой итеративной процедуры позволяет осуществить выбор оптимального состава и характеристик аппаратных и программных средств наиболее эффективным способом. В таблице 7 представлены результаты пяти итераций имитационного моделирования, в ходе которых был определен оптимальный состав используемых аппаратных средств. Графики, представленные на рисунке 9, иллюстрируют связи характеристик оперативности и производительности системы с составом и характеристиками входящих в нее компонентов в рамках этих итераций.

Как показали проведенные эксперименты, на определенном этапе моделирования можно столкнуться с ситуацией, когда увеличение аппаратной составляющей становится малопродуктивным. В этом случае необходимо рассмотреть возможность доработки

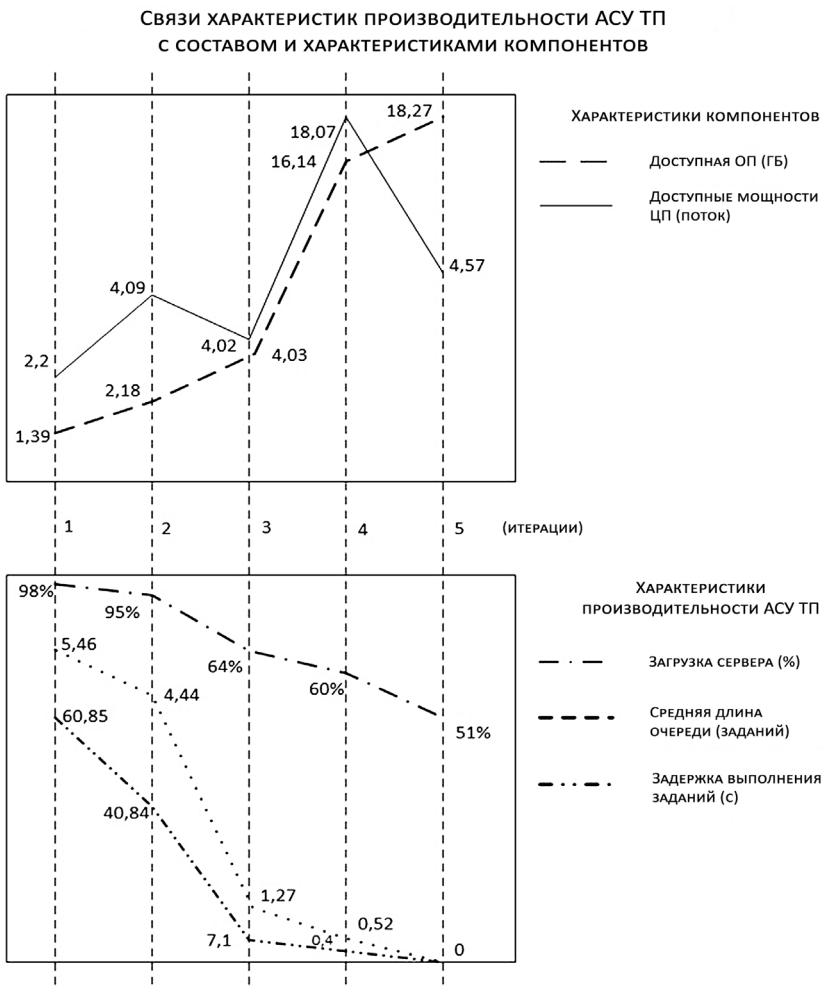

Рис. 9. Графики, иллюстрирующие связь характеристик оперативности и производительности системы с составом и характеристиками входящих в нее компонентов

Таблица 7

Результаты моделирования в среде CPN Tools

Характеристика	**1**	**2**	**3**	**4**	**5**
Задержка выполнения заданий (с)	60,85	40,84	7,10	0,40	0,00
Средняя длина очереди	5,46	4,44	1,27	0,52	0,00
Доступные мощности ЦП (потоки)	2,20	4,09	4,02	8,07	4,57
Доступная оперативная память (Гб)	1,39	2,18	4,03	16,14	18,27
Загрузка сервера в %	98	95	64	60	51

программных компонентов в части более эффективного использования аппаратных и программных ресурсов.

Надежность и точность получаемых с помощью методов и средств имитационного моделирования результатов в значительной мере зависит от адекватности и корректности моделей исследуемых систем. Важным условием построения таких моделей является определение «золотой середины» между простотой и точностью модели. Если модель слишком упрощена и в ней не учтены некоторые существенные факторы, то высока вероятность получения недостоверных результатов. С другой стороны, если модель сложная и в нее включены факторы, имеющие незначительное влияние на изучаемую систему, то резко повышаются затраты на создание ее модели и возрастают риски внесения ошибок.

В связи с этим основным направлением совершенствования представленных в статье методов и средств является разработка моделей механизмов и средств планирования и управления вычислительными процессами и ресурсами в современных аппаратно- программных комплексах и системах.

Заключение

Представленное в статье методическое, модельное и программное обеспечение представляют собой необходимый и достаточный арсенал технологий и средств для автоматизированного построения, верификации и валидации технических решений при разработке АСУ ТП, обладающих требуемым уровнем производительности и масштабируемости. Достоинством разработанных моделей и средств является предоставление возможности формализованного описания требований к таким ЭТХ АСУ ТП, как производительность, оперативность, масштабируемость и др., что позволяет реализовать эффективную процедуру имитационного математического моделирования и оценивания качества функционирования АСУ ТП на этапе ее проектирования.

Моделирование осуществляется посредством генерации рабочей нагрузки, симуляции процессов планирования и применения ресурсов, мониторинга и анализа характеристик систем и эффективности использования ресурсов. С помощью моделирования могут быть реализованы следующие виды анализа проектных решений:

- анализ чувствительности, который исследует пространство параметров для определения оптимальных рабочих параметров для штатных режимов функционирования;
- исследование альтернативных сценариев, платформ, вариантов физического развертывания и конфигураций;
- анализ масштабируемости и пропускной способности, который исследует возможности проекта или конфигурации в случае увеличения нагрузок;
- выявление опасных для функционирования системы рабочих нагрузок;
- анализ эффективности использования аппаратных и программных ресурсов.

Направлениями дальнейших исследований и работ являются программная реализация и отладка представленных в статье моделей и методов.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон от 26 июля 2017 г. N 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации». URL: https://rg.ru/documents/2017/07/31/bezopasnost-dok.html (дата обращения: 15.05.2023).

2. ГОСТ Р 27.102-2021. Национальный стандарт Российской Федерации. Надежность в технике. Надежность объекта. Термины и определения. URL: https://docs.cntd.ru/document/1200181141 (дата обращения: 15.05.2023).

3. Rational Unified Process. Best Practices for Software. Development Teams // Rational Software White Paper TP026B, Rev 11/01. URL: http://www.protesting.ru/documentation/RUP_bestpractices_TP026B.pdf (дата обращения: 15.05.2023).

4. Методологии разработки ПО: Microsoft Solutions Framework. URL: https://gb.ru/posts/development_methodology_msf (дата обращения: 15.05.2023).

5. Федорова О.В., Мамаева А.А., Якунина Е.А. Применение методологий SADT и ARIS для моделирования и управления бизнес-процессами информационных систем // Вестник Воронежского государственного университета инженерных технологий. 2018. Т. 80, № 1. С. 105–109. URL: https://doi.org/10.20914/2310-1202-2018-1-105-109 (дата обращения: 15.05.2023).

6. UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems. OMG Document Number: formal/2009-11-02. URL: http://www.omg.org/spec/MARTE/1.0 (дата обращения: 15.05.2023).

7. Баев А.В., Самонов А.В., Сафонов В.М. Методика проектирования автоматизированных систем управления специальными организационно-техническими системами // Моделирование, оптимизация и информационные технологии. 2021. Т. 9, № 4 (35). С. 165–182.

8. Тарасюк И.В., Масиа С.Х., Валеро Р.В. Анализ производительности параллельных систем в алгебре dtsiPBC // Программирование. 2014. Т. 40, № 5. С. 3–27.

9. Stochastic Process Algebras / A. Clark, St. Gilmore, J. Hillston, M. Tribastone. URL: https://www.dcs.ed.ac.uk/pepa/stochasticprocessalgebras.pdf (дата обращения: 15.05.2023).

10. Кирьянчиков В.А. Анализ эффективности и надежности вычислительных систем на основе стохастических сетей Петри // Известия СПбГЭТУ «ЛЭТИ». 2020. № 8–9. С. 5–10.

11. Shailesh T., Nayak A., Prasad D. An UML Based Performance Evaluation of Real-Time Systems Using Timed Petri Net. URL: https://www.mdpi.com/2073-431X/9/4/94 (дата обращения: 15.05.2023).

12. Thong W.J., Ameedeen M.A. A Survey of Petri Net Tools // Lecture Notes in Electrical Engineering. 2015.

Vol. 315. pp. 537–551. DOI: 10.1007/978-3-319-07674-4_51.

13. Smart selection from petri net modeling tools for fast developing a manufacturing system / Yi-Nan Lin, Cheng-Ying Yang, Gwo-Jen Chiou et all. // Cogent Engineering. 2022. Vol. 9, iss. 1. 2020609. DOI: 10.1080/23311916.2021.2020609.

14. Petri Net based modeling and analysis for improved resource utilization in cloud computing / Ali M. Rizwan, F. Ahmad, Chaudary M. Hasanain, Khan Z. Ashfaq, M.A. Alqahtani, Alqurni J. Saad, Z. Ullah, W.U. Khan // PeerJ Comput. Sci. 2021. 7:e351. DOI: 10.7717/peerj-cs.351.

15. Максимей И.В., Смородин В.С., Демиденко О.М. Разработка имитационных моделей сложных технических систем : монография / М-во образования РБ, Гомельский гос. ун-т им. Ф. Скорины. Гомель : ГГУ им. Ф. Скорины, 2014. 298 с.

16. Аюпов В.В. Математическое моделирование технических систем : учеб. пособие / Перм. гос. с.-х. акад. им. Д. Н. Прянишникова. Пермь : ПрокростЪ, 2017. 242 с.

REFERENCES

1. *Federalnyi zakon ot 26 iiulia 2017 g. N 187-FZ "O bezopasnosti kriticheskoi informatsionnoi infrastruktury Rossiiskoi Federatsii"* [Federal Law of the Russian Federation No. 187-FZ on the Security of Critical Information Infrastructure of the Russian Federation, dtd. July 26, 2017]. Available at: https://rg.ru/documents/2017/07/31/bezopasnost-dok.html (accessed 15.05.2023).

2. *GOST R 27.102–2021. Natsionalnyi standart Rossiiskoi Federatsii. Nadezhnost v tekhnike. Nadezhnost obekta. Terminy i opredeleniia* [National Standard of the Russian Federation GOST R 27.102–2021. Dependability in Technics. Dependability of Item. Terms and Definitions]. Available at: https://docs.cntd.ru/document/1200181141 (accessed 15.05.2023).

3. Rational Unified Process. Best Practices for Software Development Teams. *Rational Software White Paper TP026B*, Rev 11/01. Available at: http://www.protesting.ru/documentation/RUP_bestpractices_TP026B.pdf (accessed 15.05.2023).

4. *Metodologii razrabotki PO: Microsoft Solutions Framework* [Software Development Methodologies: Microsoft Solutions Framework]. Available at: https://gb.ru/posts/development_methodology_msf (accessed 15.05.2023).

5. Fedorova O.V., Mamaeva A.A., Yakunina E.A. Primenenie metodologii SADT i ARIS dlia modelirovaniia i upravleniia biznes-protsessami informatsionnykh system [Application of SADT and ARIS Methodologies for Modeling and Management of Business Processes of Information Systems]. *Vestnik Voronezhskogo gosudarstvennogo universiteta inzhenernykh tekhnologii* [Proceedings of the Voronezh State University of Engineering Technologies], 2018, vol. 80, no. 1, pp. 105–109. Available at: https://doi.org/10.20914/2310-1202-2018-1-105-109 (accessed: 15.05.2023).

6. *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems.* OMG Document Number: formal/2009-11-02. Available at: http://www.omg.org/spec/MARTE/1.0 (accessed 15.05.2023).

7. Baev A.V., Samonov A.V., Safonov V.M. Metodika proektirovaniia avtomatizirovannykh sistem upravleniia spetsialnymi organizatsionno-tekhnicheskimi sistemami [Methodology of Designing Automated Control Systems for Special Organizational and Technical Systems]. *Modelirovanie, optimizatsiia i informatsionnye tekhnologii* [Modeling, Optimzation, and Information Technologies], 2021, vol. 9, no. 4 (35), pp. 165–182.

8. Tarasiuk I.V., Macia S.Kh., Valero R.V. Analiz proizvoditelnosti parallelnykh sistem v algebre dtsiPBC [Performance Analysis of Concurrent Systems in Algebra dtsiPBC]. *Programmirovanie* [Programming and Computer Software Journal], 2014, vol. 40, no. 5, pp. 3–27.

9. Clark A., Gilmore St., Hillston J., Tribastone M. *Stochastic Process Algebras*. Available at: https://www.dcs.ed.ac.uk/pepa/stochasticprocessalgebras.pdf (accessed: 15.05.2023).

10. Kiranchikov V.A. Analiz effektivnosti i nadezhnosti vychislitelnykh sistem na osnove stokhasticheskikh setei Petri [The Efficiency and Reliability Analysis of Computer Systems based on Stochastic Petri Nets]. *Izvestiia SPbGETU "LETI"* [Proceedings of Saint Petersburg Electrotechnical University], 2020, no. 8–9, pp. 5–10.

11. Shailesh T., Nayak A., Prasad D. *An UML Based Performance Evaluation of Real-Time Systems Using Timed Petri Net.* Available at: https://www.mdpi.com/2073-431X/9/4/94 (accessed 15.05.2023).

12. Thong W.J., Ameedeen M.A. A Survey of Petri Net Tools. *Lecture Notes in Electrical Engineering*, 2015, vol. 315, pp. 537–551. DOI: 10.1007/978-3-319-07674-4_51.

13. Yi-Nan Lin, Cheng-Ying Yang, Gwo-Jen Chiou et all. Smart Selection from Petri Net Modeling Tools for Fast Developing a Manufacturing System. *Cogent Engineering*, 2022, vol. 9, iss. 1, 2020609. DOI: 10.1080/23311916.2021.2020609.

14. Ali M. Rizwan, F. Ahmad, Chaudary M. Hasanain, Khan Z. Ashfaq, M.A. Alqahtani, Alqurni J. Saad, Z. Ullah, W.U. Khan Petri Net based Modeling and Analysis for Improved Resource Utilization in Cloud Computing. *PeerJ Comput. Sci.*, 2021, 7:e351. DOI: 10.7717/peerj-cs.351.

15. Maksimei I.V., Smorodin V.S., Demidenko O.M. *Razrabotka imitatsionnykh modelei slozhnykh tekhnicheskikh system.* Monografiia M-vo obrazovaniia RB, Gomelskiy gos. un-t im. F. Skoriny [The Development of Simulation Models for Sophisticated Engineering Systems. Monography prepared at the Skoriny Gomel State University, the Ministry of Education of the Republic of Belarus]. Gomel, Skoriny Gomel State University Publ., 2014. 298 p.

16. Ayupov V.V. *Matematicheskoe modelirovanie tekhnicheskikh system.* Ucheb. posobie. Perm. gos. s.-kh. akad. im. D. N. Pryanishnikova [Mathematical Modeling of Engineering Systems. Textbook issued in Pryanishnikov Perm State Agricultural Academy]. Perm, Prokrost Publ., 2017. 242 p.