

На правах рукописи



Марченков Сергей Александрович

**АВТОМАТИЗИРОВАННАЯ РАЗРАБОТКА ИНТЕРОПЕРАБЕЛЬНОЙ
ПРОГРАММНОЙ ИНФРАСТРУКТУРЫ ДЛЯ ОРГАНИЗАЦИИ
СОВМЕСТНО ИСПОЛЬЗУЕМОГО ИНФОРМАЦИОННОГО
ИНТЕРНЕТ-ОКРУЖЕНИЯ**

Специальность 05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и
компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Петрозаводск – 2019

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего образования «Петрозаводский государственный университет» (ПетрГУ).

Научный руководитель: кандидат физико-математических наук, доцент
Корзун Дмитрий Жоржевич
Федеральное государственное бюджетное образовательное учреждение высшего образования «Петрозаводский государственный университет» (ПетрГУ), доцент кафедры информатики и математического обеспечения

Официальные оппоненты: доктор физико-математических наук, профессор
Терехов Андрей Николаевич
Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет» (СПбГУ), заведующий кафедрой системного программирования

кандидат технических наук
Кринкин Кирилл Владимирович
Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)» (СПбГЭТУ «ЛЭТИ»), доцент, заведующий кафедрой математического обеспечения и применения ЭВМ

Ведущая организация: Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М. В. Ломоносова» (МГУ)

Защита состоится «__» _____ 2019 г. в ____ часов на заседании диссертационного совета Д 002.199.01 при Федеральном государственном бюджетном учреждении науки Санкт-Петербургском институте информатики и автоматизации Российской академии наук (СПИИРАН) по адресу: 199178, Санкт-Петербург, 14 линия В.О., 39, ауд. 401. Факс: (812)-328- 44-50 Тел: (812)-328-34-11

С диссертацией можно ознакомиться в библиотеке Федерального государственного бюджетного учреждения науки Санкт-Петербургского института информатики и автоматизации Российской академии наук по адресу: 199178, Санкт-Петербург, В.О., 14 линия, д. 39 и на сайте <http://www.spiiras.nw.ru/dissovet/>.

Автореферат разослан «__» _____ 2019 г.

Ученый секретарь
диссертационного совета Д 002.199.01,
кандидат технических наук



Зайцева А.А.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Совместно используемое информационное интернет-окружение (СИИО) поддерживает обмен информацией в коллективе людей при решении ими общей задачи из некоторой предметной области. Человеку предоставляются цифровые сервисы, ускоряющие информационную поддержку (обмен и анализ) за счет использования доступных в СИИО ресурсов, в том числе и экспертных ресурсов самого человека. Развитие технологий СИИО и их применение формируют актуальное направление процесса цифровизации различных сфер жизни общества, включая экономику, науку, образование, культуру. В СИИО интегрируются информационные, технические и экспертные ресурсы, организуется совместный доступ к общей информации, обеспечивается контекстная осведомленность человека. Для интерактивной мультимедийной информационной поддержки и видеоконференцсвязи по обмену экспертными знаниями используются широкоформатные экраны, аудиосистемы помещения, персональные мобильные устройства пользователей. Для обработки данных и извлечения из них информации используется локальное вычислительное и сетевое оборудование в соответствии с подходами периферийных и туманных вычислений. Ресурсы являются неоднородными и динамичными, что затрудняет их интеграцию при построении сервиса и увеличивает трудозатраты на разработку программного обеспечения (ПО). Областью данного исследования выступает автоматизированная разработка ПО, выполняющего интеграцию ресурсов при построении сервиса в СИИО.

На методы разработки ПО для СИИО влияет концепция окружающего интеллекта и соответствующих технологий искусственного интеллекта (ИИ). Возрастает роль средств автоматизированной разработки ПО для использования технологий ИИ. Информационная поддержка пользователей усиливается за счет контекстных сервисов, учитывающих состояние пользователя и имеющиеся ресурсы. Сервисы приобретают свойства «интеллектуальности»: распознавание текущей ситуации, анализ поведения человека, предоставление информации в ненавязчивой форме. Для реализации этих свойств используют известные методы многоагентных систем и технологии Семантического веба, что приводит к следующим проблемам: а) нет достаточного (для имеющегося разнообразия ресурсов) метода разработки ПО с интеграцией динамичных и неоднородных ресурсов в СИИО; б) нет достаточных (для имеющегося разнообразия предметных областей) моделей и шаблонов проектирования контекстных сервисов, где сервис строится как многоагентная система с информационно-управляемым взаимодействием на основе технологий Семантического веба; в) нет развитых (с существенной долей кодогенерации) средств программирования взаимодействия агентов, требуемого для построения сервиса.

В диссертационной работе предлагается использовать подход интеллектуальных пространств (ИП) для создания СИИО. Построение сервиса выполняется программными агентами, каждый отвечает за определенный ресурс в СИИО. В соответствии с подходом ИП, агенты создают и поддерживают общее информационное содержимое для семантического связывания ресурсов. В нем представлены виртуальные цифровые образы (двойники) имеющихся ресурсов, пользователей и происходящих процессов совместного использования информации. Управление информационным содержимым основано на интероперабельной программной инфраструктуре, отвечающей за организацию информационно-управляемого взаимодействия агентов. Существующие методы разработки программной инфраструктуры приводят к значительным трудозатратам на создание и сопровождение ПО. Необходим метод автоматизированной разработки программной инфраструктуры, обеспечивающий семантическую интероперабельность агентов при построении ими сервисов на основе интеграции ресурсов в СИИО.

Степень разработанности темы. Исследование вопросов создания СИИО ведется в России и за рубежом. Область является междисциплинарной – изучаются свойства информационного обмена и оперативного анализа информации человеком с целью создания и применения цифровых технологий информационной поддержки деятельности людей в различных предметных областях. Развитию методологических основ разработки ПО для технологий информационной сервисно-ориентированной поддержки способствовали труды российских и зарубежных ученых: С.И. Баландина, В.А. Васенина, В.И. Городецкого, А.В. Гуртова, А.М. Кашевника, Ю. Кильяндера, Д.Ж. Корзуна, К.В. Кринкина, Д. Кук, Д.Е. Намиота, Я. Оливера, А.Л. Ронжина, А.В. Смирнова, В.Б. Тарасова, А.Н. Терехова, Р.М. Юсупова, Т. Чинотти и др.

Целью диссертационной работы является повышение эффективности разработки программной инфраструктуры совместно используемого информационного интернет-окружения за счет: а) унифицированного моделирования сервиса как системы взаимодействующих агентов и б) автоматизированного программирования взаимодействия на основе кодогенерации. Для достижения этой цели исследуются следующие **задачи**:

1. Выполнить анализ результатов современных исследований в области информационного обмена и анализа. Провести обзор существующих решений для совместно используемых информационных интернет-окружений, а также моделей проектирования сервисов, алгоритмов автоматизации программирования агентов и методов разработки программных инфраструктур. Сформулировать основные требования к разработке ПО, выполняющего интеграцию ресурсов.

2. Сформировать метод разработки интероперабельной программной инфраструктуры СИИО, обеспечивающей информационно-управляемое взаимодействие агентов для интеграции динамических и неоднородных ресурсов при построении сервиса.

3. Предложить концептуальную модель сервиса СИИО, позволяющую описывать с помощью онтологии варианты информационно-управляемого взаимодействия агентов для построения контекстных сервисов и их композиции на основе технологий Семантического веба.

4. Разработать алгоритм автоматизации программирования взаимодействия агентов СИИО, обеспечивающего кодогенерацию программных механизмов информационно-управляемого взаимодействия по заданной спецификации проектирования сервиса.

5. Разработать набор предметно-ориентированных моделей проектирования сервисов СИИО в качестве шаблонных решений для востребованных приложений.

6. Реализовать комплекс программных средств в составе: а) генератор программного кода взаимодействия агентов на основе полученных алгоритма автоматизации программирования и концептуальной модели сервиса; б) экспериментальные образцы предметно-ориентированных сервисов для вычислительных сред интеллектуального зала, умного музея и промышленного предприятия. Выполнить экспериментальное исследование на основе комплекса программных средств с целью оценки эффективности сформированного метода разработки программной инфраструктуры СИИО.

Объектом исследования является интероперабельная программная инфраструктура, обеспечивающая организацию информационно-управляемого взаимодействия агентов при построении ими сервисов на основе интеграции ресурсов вычислительной среды.

Предметом исследования являются модели проектирования сервисов как много-агентных систем, использующих технологии Семантического веба для информационного обмена, и алгоритмы автоматизации программирования информационно-управляемого взаимодействия, обеспечивающие кодогенерацию для реализации программ агентов.

Научная новизна диссертационной работы состоит в следующем:

1. Предложен метод разработки интероперабельной программной инфраструктуры СИИО, отличающийся возможностью унифицированной и автоматизированной разработ-

ки сервиса как системы с информационно-управляемым взаимодействием агентов для интеграции динамических и неоднородных ресурсов при построении сервиса.

2. Предложена концептуальная модель информационного сервиса СИИО, отличающаяся возможностью онтологического описания информационно-управляемого взаимодействия агентов для построения контекстных сервисов и их композиции на основе технологий Семантического веба.

3. Предложен алгоритм автоматизации программирования взаимодействия агентов СИИО, отличающийся возможностью кодогенерации программных механизмов информационно-управляемого взаимодействия для построения сервиса, в дополнение к структурам данных предметной области.

4. Предложен набор предметно-ориентированных моделей проектирования сервисов СИИО, отличающихся предоставлением разработчику архитектурных и поведенческих абстракций информационно-управляемого взаимодействия агентов как шаблонных решений для востребованных приложений СИИО в составе: а) распознавание присутствия и анализ активности пользователей, б) сопровождение и визуализация плана деятельности людей, в) совместное пополнение информационного содержимого знаниями о предметной области, г) мониторинг объектов физической среды.

5. Разработан комплекс программных средств в соответствии с предложенными новым методом разработки программной инфраструктуры, моделями проектирования сервисов и алгоритмом автоматизации программирования взаимодействия агентов.

Теоретическая и практическая значимость. Предложенный в работе метод разработки развивает научные основы построения многоагентных информационных систем в условиях периферийных и туманных вычислений за счет введения новых моделей проектирования сервисов и алгоритма автоматизации программирования взаимодействия агентов. Достигается снижение трудозатрат на разработку и сопровождение сервисов СИИО. Реализован генератор программного кода участвующих в построении сервисов взаимодействующих агентов. Получены экспериментальные образцы ПО и оценки их эффективности, допускающие практическое использование при разработке СИИО для вычислительных сред интеллектуального зала, умного музея и промышленного предприятия.

Методы исследования. Результаты выполненных прикладных исследований и разработок основаны на методах программирования распределенных и многоагентных систем, методах онтологического моделирования и Семантического веба, технологиях интеллектуальных пространств и интеллектуальных Интернет-технологиях.

Основные положения, выносимые на защиту.

1. Метод разработки интероперабельной программной инфраструктуры СИИО для унифицированной и автоматизированной разработки сервиса как многоагентной системы, интегрирующей динамические и неоднородные ресурсы при построении сервиса.

2. Концептуальная модель информационного сервиса СИИО для проектирования вычислительного процесса построения контекстных сервисов и их композиции на основе технологий Семантического веба с информационно-управляемым взаимодействием агентов.

3. Алгоритм автоматизации программирования взаимодействия агентов СИИО для кодогенерации программных механизмов информационно-управляемого взаимодействия по заданной онтологии информационного сервиса.

4. Предметно-ориентированные модели проектирования сервисов для разработки СИИО на основе шаблонных решений для востребованных приложений: а) распознавание присутствия и анализ активности пользователей, б) сопровождение и визуализация плана деятельности людей, в) совместное пополнение информационного содержимого знаниями о предметной области, г) мониторинг объектов физической среды.

5. Комплекс программных средств для метода разработки программной инфраструктуры СИИО в составе: а) генератор программного кода взаимодействия агентов по онтологической модели информационного сервиса; б) экспериментальные образцы ПО предметно-ориентированных сервисов для вычислительных сред интеллектуального зала, умного музея и промышленного предприятия.

Степень достоверности. Достоверность научных положений, выводов и результатов диссертационной работы обеспечивается за счет анализа состояния исследований в данной области, согласованности теоретических выводов с результатами экспериментального исследования полученной программной реализации, а также апробацией основных положений диссертации на научных конференциях, в научных работах и приравненных к ним публикациях. По результатам исследования получены 7 свидетельств о государственной регистрации программ для ЭВМ и 1 свидетельство о государственной регистрации базы данных.

Апробация результатов исследования. Результаты диссертационного исследования представлялись на международных и российских научных мероприятиях, в том числе: международные конференции ассоциации открытых инноваций FRUCT (Россия, Финляндия, 2013–2019 гг.), международная IEEE конференция «Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)» (Румыния, Бухарест, 2017), научный семинар «Проблемы современных информационно-вычислительных систем» (Россия, Москва, 2017), всероссийская конференция «Цифровые технологии в образовании, науке, обществе» (Россия, Петрозаводск, 2017–2018 гг.), конференция «Современные технологии в теории и практике программирования» (Россия, Санкт-Петербург, 2013–2016 гг.).

Реализация результатов работы. Работа выполнена по заданию № 2.5124.2017/8.9 Минобрнауки России базовой части государственного задания ПетрГУ на 2017–2019 гг. Междисциплинарное научное исследование и разработка поддержаны в рамках проектов: ОГОН РФФИ № 16-01-12033 (2016–2017 гг.); ФЦП по соглашению № 14.574.21.0060 (2014–2016 гг.); Задания № 2014/154 Минобрнауки России базовой части государственного задания (2014–2016 гг.). Работа поддержана Программой развития опорного университета для ПетрГУ на 2017–2021 гг. Полученные результаты используются в учебном процессе ПетрГУ, для производства и развития цифровых сервисов в ООО «Опти-Софт» и при разработке программно-аппаратного комплекса мониторинга роботизированного производственного оборудования.

Научные публикации. По теме диссертации опубликовано 28 научных работ и приравненных к ним публикаций, среди которых 3 работы в журналах из списка ВАК и 9 работ в международных изданиях, индексируемых в реферативных базах Web of Science и Scopus, 7 свидетельств о государственной регистрации программ для ЭВМ и 1 свидетельство о государственной регистрации базы данных.

Структура и объем работы. Диссертация объемом 155 машинописных страниц, содержит введение, четыре главы и заключение, список литературы (125 наименований), 9 таблиц, 32 рисунка и 2 приложения.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы исследования, определяется цель и решаемые задачи, формулируются положения, выносимые на защиту, их научная новизна, теоретическая и практическая значимость исследования.

В первой главе приводится концептуальная модель для содержательного описания структуры (множество понятий и связей между ними) совместно-используемого информационного интернет-окружения. Раскрывается актуальность и востребованность СИИО в развитии процессов информатизации общества (цифровая экономика, наука и образование, культура). Выполнен анализ результатов современных исследований в

области информационного обмена и анализа на основе многоагентных систем и технологий Семантического веба. В целом, в главе показано, что требуется новый метод разработки программной инфраструктуры СИИО.

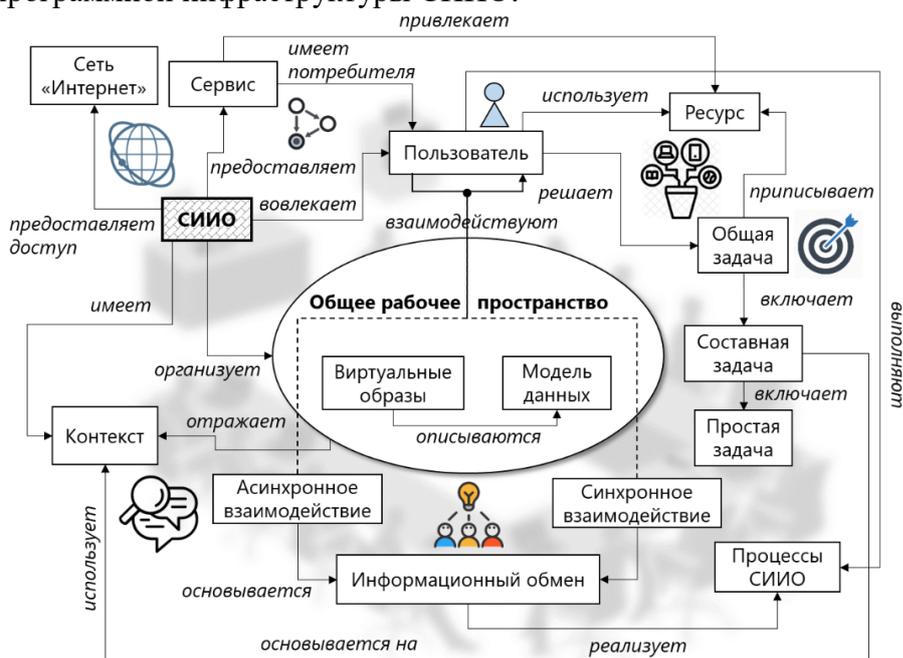


Рисунок 1 – Концептуальная модель СИИО

В соответствии с рис. 1 понятие СИИО описывается с помощью определяющих структурных понятий и отношений между ними. Как информационная система, СИИО поддерживает обмен информацией в группе людей при решении ими общей задачи некоторой предметной области. Рассмотрены характеристики сервис-ориентированного обмена и совместного анализа информации. Человеку предоставляются цифровые сервисы, ускоряющие информационную поддержку за счет интеграции доступных ресурсов: информационные, технические и экспертные ресурсы. Опорные примеры предметных областей – интеллектуальный зал (сопровождение плана деятельности людей), умный музей (сопровождение историко-культурного знания), промышленный интернет (мониторинг объектов физической среды). Показано, что требуются как новые общие модели сервиса для разработки СИИО, так и новые предметно-ориентированные модели проектирования сервисов на основе шаблонных решений для востребованных предметных областей.

Выполнен обзор основных классов программных систем, применяемых для разработки СИИО с учетом реализуемой функции (обмен сообщениями, обеспечение видео- и телеконференцсвязи, проведение совещаний и др.). Выделены основные вычислительные элементы, обеспечивающие процесс совместного использования СИИО. В программной инфраструктуре эти элементы представлены агентами, каждый отвечает за определенный ресурс. Требуется реализация взаимодействия агентов в программном коде этих агентов. Для автоматизации программирования агентов сформулированы требования к интеграции ресурсов при построении сервиса взаимодействующими агентами. Выполнен обзор существующих подходов и технологий в области исследования, чтобы оценить возможности повышения эффективности разработки для предлагаемого далее метода разработки интероперабельной программной инфраструктуры.

Во второй главе представлен метод разработки интероперабельной программной инфраструктуры СИИО. Метод обеспечивает информационно-управляемое взаимодействие агентов для интеграции динамических и неоднородных ресурсов в СИИО при построении сервиса с поддержкой автоматизации программирования взаимодействия агентов (рис. 2).

Метод требует применения концептуальной модели M_{srv} информационного сервиса для проектирования сервиса как системы взаимодействующих агентов. Модель дает уни-

фицированное онтологическое описание процесса построения сервиса с учетом контекста, взаимодействующих агентов и интегрируемых ресурсов. Онтологическое описание сервиса использует выделенные категории сервисов и типовые модели информационно-управляемого взаимодействия агентов.

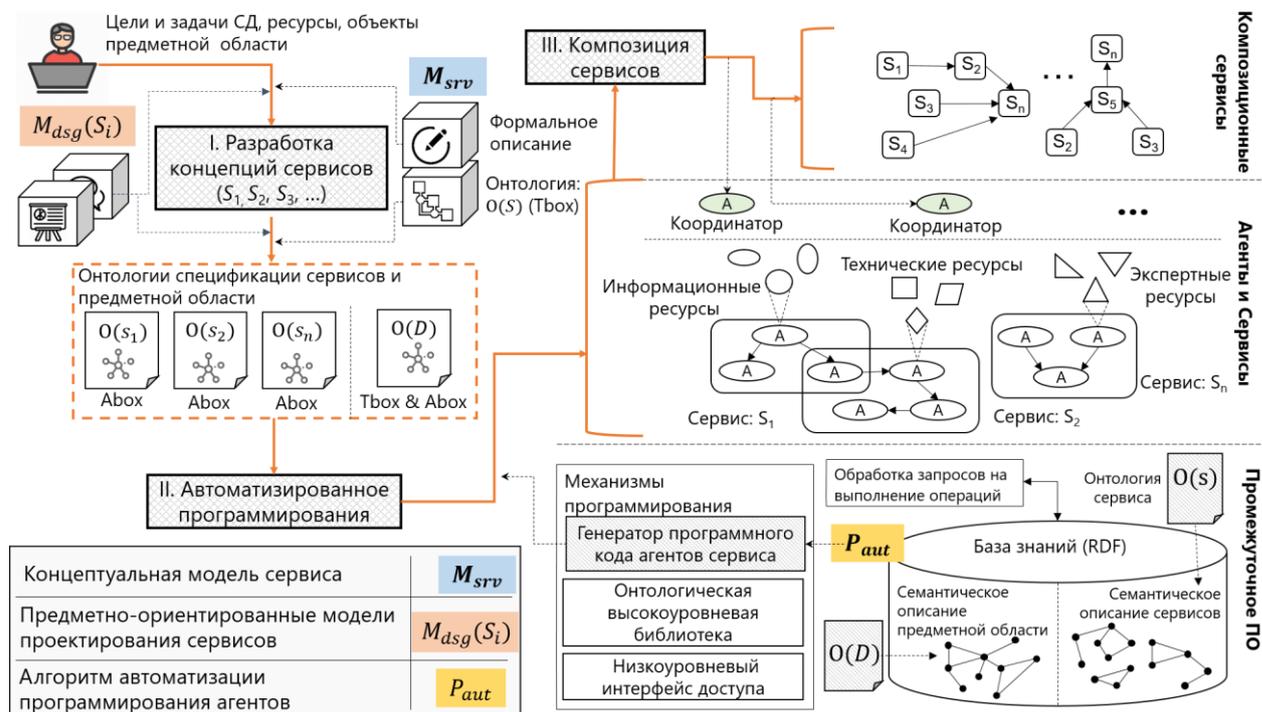


Рисунок 2 – Общая схема метода разработки программной инфраструктуры СИИО

Метод поддерживает применение предметно-ориентированных моделей проектирования $M_{dsg}(S_i)$ при разработке сервиса для конкретной предметной области: а) распознавание присутствия и анализ активности пользователей, б) сопровождение и визуализация плана деятельности людей, в) совместное пополнение информационного содержимого знаниями о предметной области, г) мониторинг объектов физической среды. Эти модели предлагают готовые архитектурные и поведенческие абстракции взаимодействия агентов, полученных на основе концептуальной модели.

В качестве инструментального средства метод включает алгоритм автоматизации программирования взаимодействия агентов P_{aut} , позволяя по заданным моделям проектирования M_{srv} и $M_{dsg}(S_i)$ генерировать программный код интероперабельных структур данных и реализации сценариев взаимодействия агентов.

Метод разбивает разработку программной инфраструктуры на следующие этапы: а) разработка моделей проектирования поддерживаемых сервисов, б) автоматизированное программирование взаимодействия агентов при построении сервисов, в) композиция контекстных сервисов для интеграции ресурсов.

Концептуальная модель информационного сервиса СИИО дает концептуальное описание сервиса с использованием категорий сервисов и моделей организации информационно-управляемого взаимодействия агентов в соответствии с технологиями интеллектуальных пространств. Сервис представлен как система взаимодействующих (информационно-управляемое взаимодействие) агентов для реализации предписанной функции с интеграцией ресурсов. Доступ к системе предоставляется с помощью описанного заранее интерфейса и осуществляется в соответствии с ограничениями и политиками, определяемыми в описании сервиса, а построение сервиса следует заданному набору действий по взаимодействию агентов.

Формально, сервис определяется на основе совокупности следующих элементов:

$$S^k(r) = \langle \{A_t^c\}_{t=1}^n, M(k), I_{pr}, I_{rs} \rangle,$$

где $r \in \mathcal{R}$ – задействованный ресурс r из множества доступных ресурсов \mathcal{R} ; $k \in \mathcal{K}$ – категория сервисов k из множества категорий \mathcal{K} ; $\{A_t^r\}_{t=1}^n$ – агенты, участвующие в построении сервиса (r – роль агента, n – количество агентов, t – порядковый номер агента); $c \in \mathcal{C}$ – роль агента c из множества \mathcal{C} ; $M(k)$ – модель взаимодействия агентов (определяет протокол взаимодействия для каждого агента, архитектурные абстракции и типовую модель взаимодействия); I_{pr} – начальное состояние информационного содержимого; I_{rs} – результирующее состояние информационного содержимого. На рис. 3 представлена общая схема концептуальной модели информационного сервиса.

Предлагается использовать стандарты веб-сервисов. В сочетании с методами Семантического веба такие сервисы определяются как семантические, т.е. имеют однозначно описанную семантику. Предлагаемая концептуальная модель сервиса дает унифицированное описание семантики сервиса: интерфейс (назначение сервиса, входные и выходные данные и т.д.) и происходящие процессы построения. Семантика также описывает информационно-управляемое взаимодействие агентов при построении сервиса.

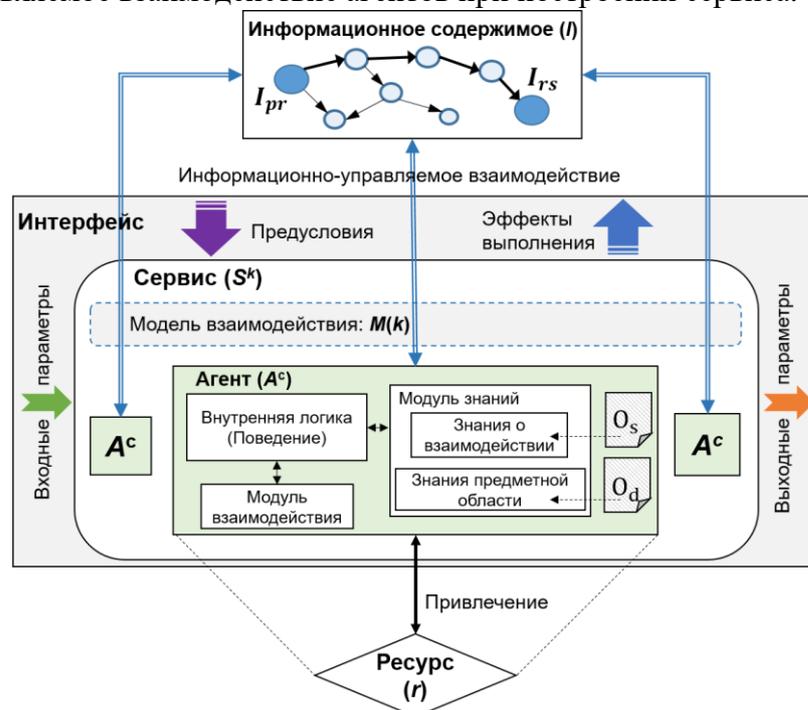


Рисунок 3 – Общая схема концептуальной модели информационного сервиса

Основным назначением концептуальной модели является разработка унифицированного онтологического описания сервиса. Предлагается онтология сервиса для СИИО (рис. 4), расширяющая онтологию OWL-S. Профиль сервиса в онтологии описывает сервис с помощью модели IOPEs (Inputs, Outputs, Preconditions Effects): входные параметры, выходные параметры, предварительные условия, эффекты/результаты выполнения.

Начальное состояние информационного содержимого для инициализации построения сервиса задается с помощью предварительного условия (*has_precondition*), а желаемое состояние информационного содержимого определяется с помощью результирующего условия (*has_result*). Предусловия и результаты задаются с помощью логических выражений. Логические выражения представлены как литералы (literals) в SPARQL-выражениях, определяемых с помощью класса *SPARQL-Expression*.

Свойство *has_category* описывает категории сервисов (по предложенной категоризации сервисов). Категория определяет типовую модель информационно-управляемого вза-

имодействия агентов: выполняемое воздействие, основополагающий ресурс, способ интеграции, выполняемая функция совместного использования.

Онтология позволяет учесть приоритеты и предпочтения пользователя, обеспечивая персонализацию сервиса. Существующая онтология профиля пользователя (FOAF) расширяется свойством *has_consumer* и объектом класса *Profile*, который описывается в терминах пользовательской контекстной информации: состояние окружения, где находится пользователь (*has_context*), интересы пользователей и другие аспекты личности (часть онтологии FOAF, свойство *has_personinformation*).

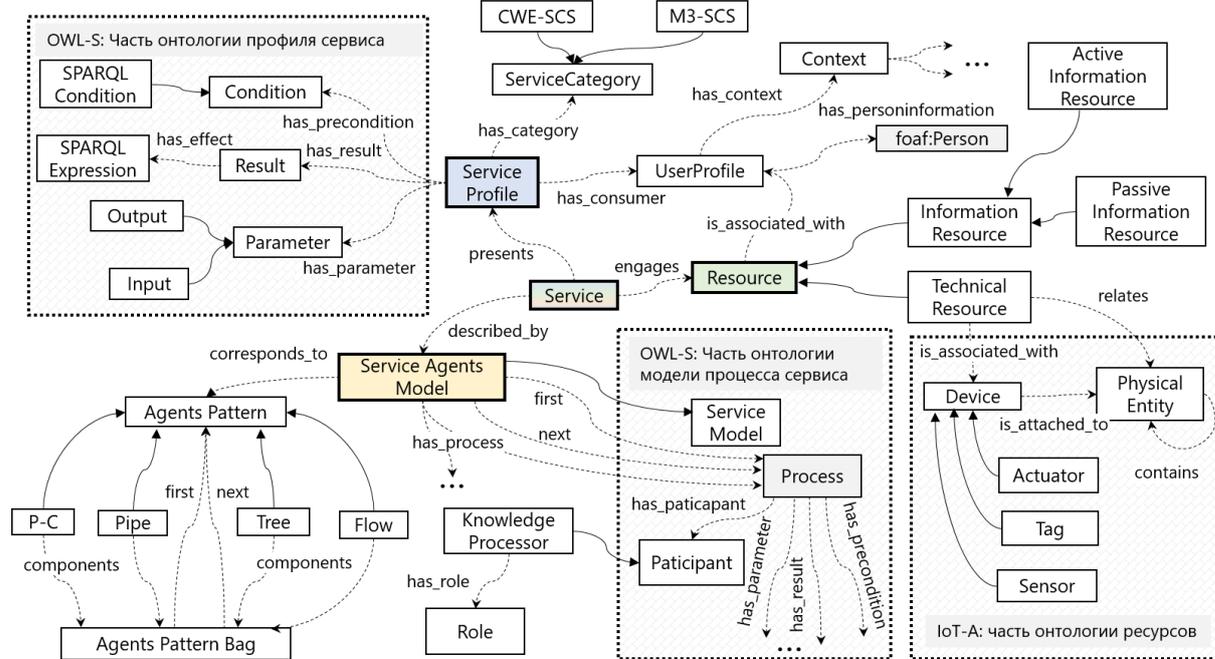


Рисунок 4 – Онтология сервиса для СИИО

Модель построения сервиса расширяется за счет класса *ServiceAgentsModel*. Взаимодействие каждого агента (*KnowledgeProcessor*) определяется его функциональной ролью (свойство *has_role*). Роль агента позволяет определить общие программные механизмы действий агента и механизмы взаимодействия с другими агентами. Взаимодействие агентов при построении сервиса происходит на основе шаблонов взаимодействия (*AgentsPattern*), описанных с помощью таких архитектурных абстракций, как *Поставщик-потребителей (P-C)*, *Конвейер (Pipe)*, *Дерево (Tree)*, *Поток (Flow)*. Взаимодействие агентов может определяться как совокупность архитектурных абстракций, представленных в некоторой последовательности (*AgentsPatternBag*).

Алгоритм автоматизации программирования взаимодействия агентов СИИО выполняет кодогенерацию для объектно-ориентированного языка программирования с использованием онтологии предметной области и информационного сервиса (рис. 5). Алгоритм состоит из двух процедур. Процедура кодогенерации объектной модели структур данных использует онтологию предметной области для генерации объектной модели. Процедура кодогенерации сценариев взаимодействия агентов использует онтологию сервиса для генерации блоков программного кода агентов.

Онтология содержит определения концептов и отношений между ними (свойства, аспекты, параметры), тогда как объектная модель использует классы для представления объектов и функции для моделирования отношений объектов и атрибутов. Процедура кодогенерации объектной модели выполняется на основе сводных правил отображения онтологии в объектную модель. Каждый онтологический класс со своими экземплярами, свойствами и ограничениями имеет свой эквивалент в объекте языка программирования.

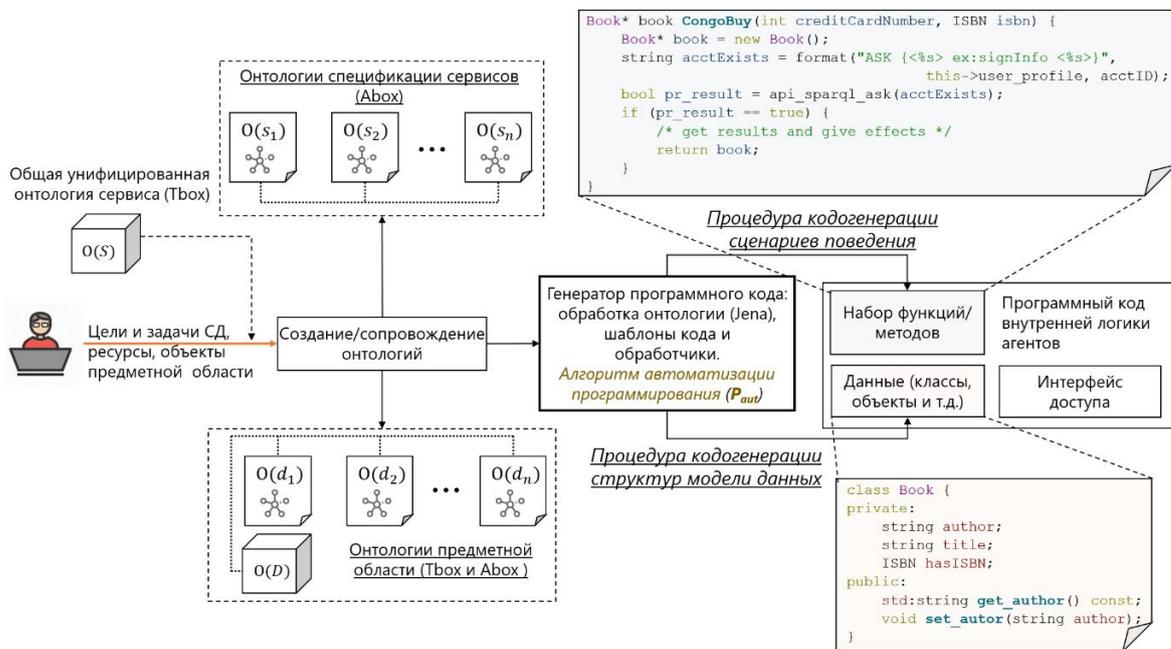


Рисунок 5 – Общая схема кодогенерации взаимодействия агентов

Построение сервиса сводится к вызову программных функций в коде агентов. Процедура кодогенерации сценариев взаимодействия агентов представлена на рис. 6. Экземпляры класса *AtomicProcess* используются для генерации кода функций. Атрибут *rdf:ID* класса *AtomicProcess* определяет имя функции. Каждое свойство *has_input* с атрибутом *rdf:ID* соответствует входным параметрам функции. Тип входного параметра может быть получен путем просмотра свойства *parameter_type*.

Внутренняя логика функций реализуется с использованием запросов SPARQL и операторов языка программирования. Свойство *has_precondition* с выражением SPARQL определяет блок кода для проверки предварительного условия, описывающего начальное состояние информационного содержимого, необходимое для инициализации построения сервиса. Генерируется блок кода, который вызывает функцию API промежуточного ПО для выполнения запроса SPARQL (как правило, запрос ASK) и проверяет результат запроса с помощью оператора «if-then-else». Аналогичный процесс генерации выполняется для блока кода, представляющего результирующее условие (*has_result*).

Свойство *has_output* со свойством *parameter_type* соответствует выходному параметру и определяет возвращаемое значение функции. XSD предоставляет необходимые типы данных (*string*, *unsignedLong* и пр.). Элементы объектной модели могут использоваться в качестве входных и выходных параметров функций. Класс *CompositeProcess* определяет функцию с вызовами внутри себя других функций, описываемых сущностями *CompositeProcess* или *AtomicProcess*. Вызовы внутренних функций задаются с помощью управляющих конструкций (*If-Then-Else*, *Repeat-While*, и т.п.), которые преобразуются в соответствующие конструкции языка программирования.

Экземпляр класса *ServiceAgentsModel* используется агентами для определения их роли в процессе построения сервисов, а также способа взаимодействия, основанного на модели «публикации/подписки». В коде каждого агента генерируется блок с необходимыми операциями подписки с использованием внутренних функций, обработчиков и функций API. Каждый запрос операции подписки задается запросом SPARQL из свойства *has_subprecondition*, используя классы и свойства предметной области. Операции подписки могут быть определены на основе результатов выполнения процессов агентов, определенных в классах *Result*. Каждый такой класс определяет изменения информационного содержимого во время построения сервиса.

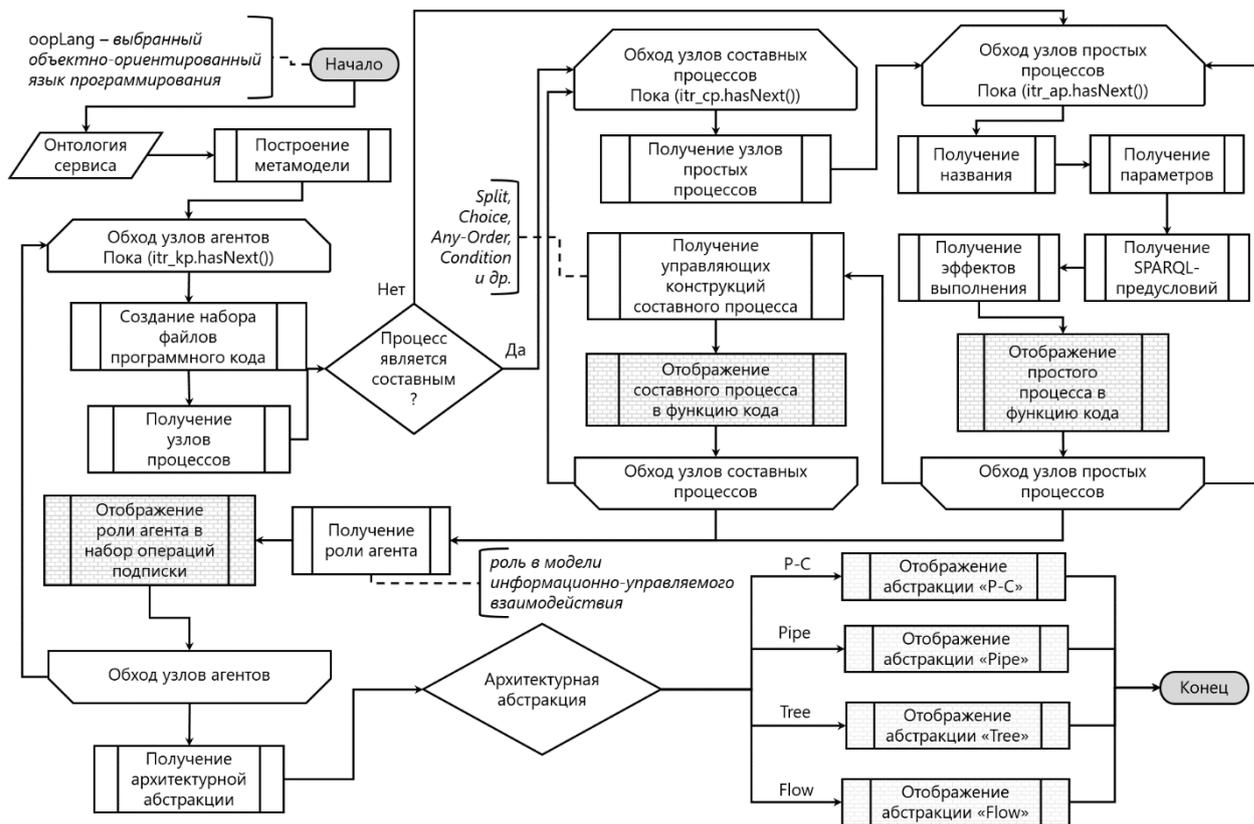


Рисунок 6 – Процедура кодогенерации сценариев взаимодействия агентов

В третьей главе представлены предметно-ориентированные модели проектирования сервисов СИИО для выполнения прикладной разработки на основе шаблонных решений для востребованных приложений. Предметно-ориентированные модели проектирования сервисов предлагаются разработчику в виде архитектурных и поведенческих моделей информационно-управляемого взаимодействия агентов для снижения трудозатрат на разработку программной инфраструктуры СИИО.

(1) Предлагается предметно-ориентированная модель проектирования для сервиса распознавания присутствия и анализа активности пользователей. Сервис позволяет собирать и использовать информацию о присутствии участников и их персональных устройств. Такая информация является исходной для построения других сервисов. Например, обнаружение прибывающих участников активирует приветственные сообщения на общественных экранах в помещении конференц-зала.

Сервис использует метод пассивной радиолокации, основанный на измерениях мощности принимаемого сигнала мобильного устройства. Мобильное устройство подключено к беспроводной локальной сети окружения. Мощность сигнала (RSSI) может быть измерена выделенным сетевым сенсором, что позволяет отслеживать, изменять и формировать информацию об уровне присутствия участника и его сетевой активности. В соответствии с рис. 7 представлена поведенческая модель информационно-управляемого взаимодействия программных агентов, участвующих в выполнении сервиса.

«Агент-адаптер сенсора» взаимодействует с выделенным сетевым сенсором, который работает на выделенном компьютере и включает в себя функциональность HTTP-сервера для получения от сенсора измерений в формате JSON. Агент обрабатывает полученные данные сенсора (временная метка, MAC-адрес, значение RSSI) и публикует их в информационное содержимое.

«Агент-агрегатор присутствия» подписан на обновление информации о присутствии мобильных устройств. Обновление сопоставляется с соответствующим пользователем (наличие в профиле информации о MAC-адресе). Агент определения присутствия уста-

навливают уровень присутствия пользователей на основе их физического присутствия в помещении СИИО и виртуального присутствия в системе (как пользователь).

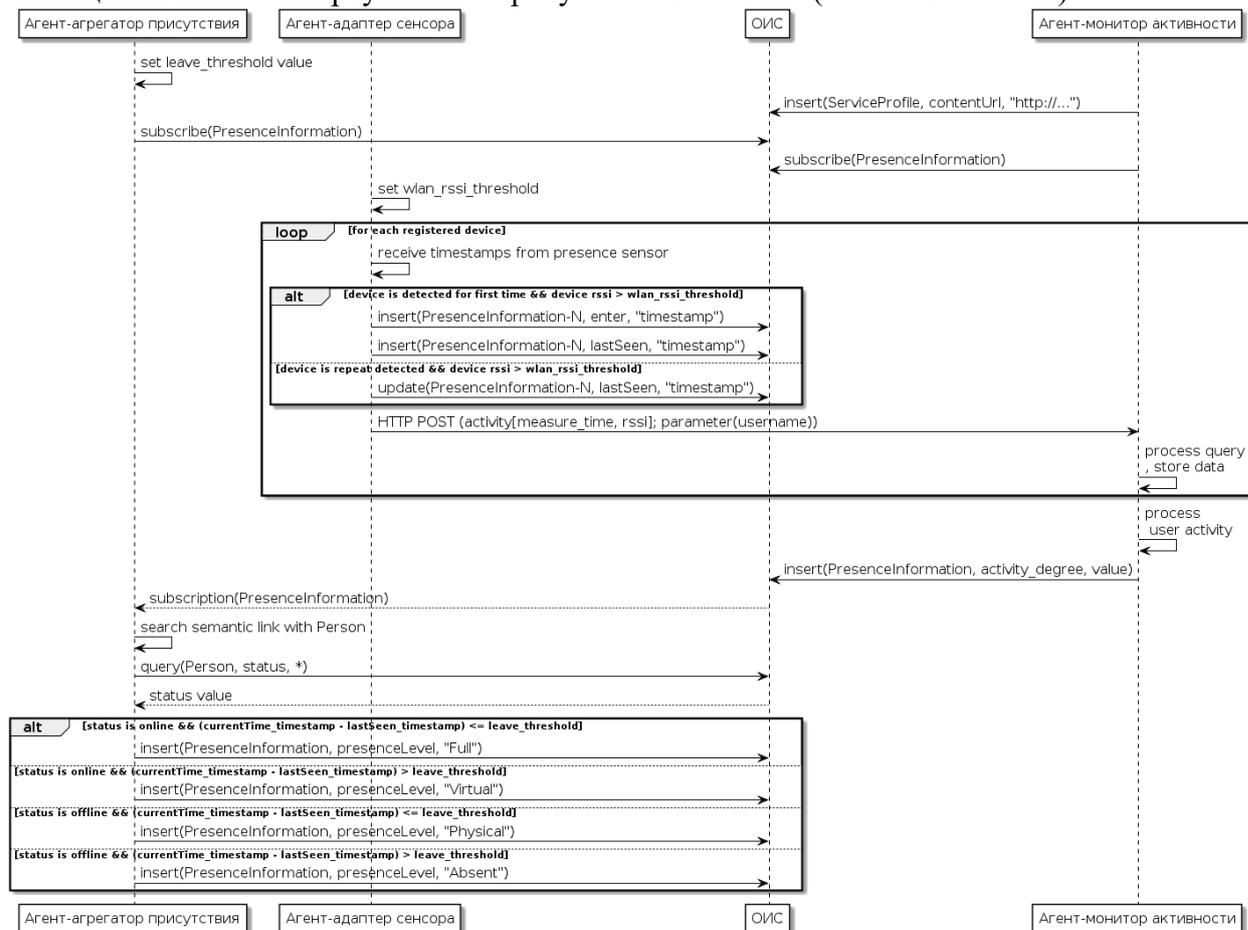


Рисунок 7 – Предметно-ориентированная модель проектирования для сервиса распознавания присутствия и анализа активности пользователей

«Агент-монитор активности» анализирует сетевую активность пользователей. Может визуализировать результаты анализа и предоставлять их другим сервисам. В конце мероприятия или по запросу агентом вычисляются суммарные показатели активности.

(2) Предлагается предметно-ориентированная модель проектирования для сервиса сопровождения и визуализации плана деятельности людей. Участвуют: агент управления программой совместной деятельности, агент визуализации программы совместной деятельности, агент визуализации медиа-информации. В онтологии сервиса класс *CollaborativeWork* имеет свойство *holdsActivity*, которое связывает его с классом *Activity*, представляющим мероприятие с названием в свойстве *activityTitle*.

В интеллектуальном зале в качестве мероприятий проводятся конференции из нескольких секций. Свойство *hasSection* связывает класс мероприятия с классом секции *Section*. Класс *Section* является основным классом для информации о секции и состоянии ее проведения. В онтологии каждый доклад представлен слотом времени (класс *Timeslot*), который хранит информацию о докладе: имя докладчика (свойство *timeslotSpeakerName*), название доклада (*timeslotTitle*), ссылка на информацию о докладчике (*timeslotPerson*), ссылка на презентацию доклада (*timeslotPresentation*), а также планируемая и фактическая длительность доклада (*timeslotDuration* и *timeslotActualDuration*). На основе даты и времени начала секции, планируемой длительности докладов строится предварительное расписание с временем начала каждого выступления. Во время мероприятия в зависимости от фактического времени начала секции и фактической продолжительности докладов выполняется пересчет времени начала следующих докладов.

(3) Предлагается *предметно-ориентированная модель проектирования для сервиса совместного пополнения информационного содержимого знаниями о предметной области*. В качестве предметной области рассматривается окружение умного музея (частный случай СИИО). Сервис обеспечивает пополнение информацией из различных источников: интернет-источники исторических данных, индивидуальная информация и исторические знания от посетителей музея, музейная информационная система. Многоагентная архитектура представлена на рис. 8. «Агент-искатель исторической информации» обеспечивает процессы и объекты деятельности людей дополнительной информацией. Агент выступает в качестве семантического посредника, выполняя преобразования информации из внешних баз данных (например, сервиса DBpedia) в семантическое представление и обратно, формируя семантическую сеть в информационном содержимом.



Рисунок 8 – Предметно-ориентированная модель проектирования для сервиса совместного пополнения информационного содержимого

Так, в умном музее пользователи могут добавлять описательную информацию об экспонатах, а также их связи с другими объектами. «Агент-агрегатор исторической информации» анализирует предоставленную информацию. «Агент-контроллер обнаружения исторических связей» выполняет связывание семантической информации на основе сопоставления узлов семантической сети (представляющих объекты, процессы и профили пользователей) друг с другом.

(4) Предлагается *предметно-ориентированная модель проектирования для сервиса мониторинга объектов физической среды*, используемого при решении производственных задач. Мониторинг состояния среды может быть организован посредством использования беспроводных сенсорных сетей (WSN). Для обеспечения интеграции и применения возможностей WSN создается отдельная инфраструктура. Основным элементом выступает сервер с базой данных для хранения информации о сенсорах и измеряемых параметрах среды. Сервис мониторинга использует сервер как источник больших данных, интегрируя информационные ресурсы в СИИО.

Сервис мониторинга реализуется следующими агентами. Агент-интерпретатор & контроллер обеспечивает информационное содержимое информацией, полученной с сервера, и выполняет динамическое обновление семантических образов WSN-узлов и связей между ними через веб-интерфейс сервера. Агент-агрегатор выполняет анализ семантических образов, извлекает знания из имеющейся информации.

В четвертой главе описывается комплекс программных средств для метода разработки программной инфраструктуры СИИО в составе: а) генератор программного кода взаимодействия агентов на основе полученного алгоритма и концептуальной модели сервиса; б) экспериментальные образцы предметно-ориентированных сервисов для

вычислительных сред интеллектуального зала, умного музея и промышленного предприятия. Выполнены экспериментальные исследования с целью оценки эффективности метода с точки зрения снижения трудозатрат на разработку и практической работоспособности получаемых программных инфраструктур.

Генератор программного кода взаимодействия агентов реализован на языке программирования Java, использует статическую схему шаблонов и обработчиков. Шаблоны кода представляют собой «предварительный код» классов, атрибутов и функций, которые реализуют элементы (классы, индивиды, свойства, ограничения) онтологии предметной области (OWL) и онтологии сервисов для СИИО (OWL+OWL-S). Представление структур данных в шаблонах с тегами трансформируется в программный код. Теги представляют собой высокоуровневые объекты, которые цепочкой преобразований переходят в конструкции целевого языка программирования. Проверку на соответствие набору логических и синтаксических соглашений, а также процедуру трансформации реализуют обработчики, заменяя специальные теги именами и элементами, извлекаемыми из онтологий. Код может быть сгенерирован для нескольких агентов, в зависимости от того, какие агенты указаны в онтологии сервиса для обеспечения его построения. Преобразование происходит при обходе RDF-графа онтологии с использованием программного каркаса Jena для построения графовой метамодели. Во время обхода генератором узлов метамодели, обработчики преобразуют шаблоны в целевой программный код (поддерживаются объектно-ориентированные языки программирования C++ и Python).

Процесс определения эффективности следует авторской методике, которая направлена на оценку трудозатрат на разработку сервисов за счет использования унифицированной онтологии сервиса и генератора программного кода взаимодействия агентов. Полученные оценки сравниваются с трудозатратами на разработку программной инфраструктуры без использования метода. Трудозатраты выражаются в человеко-часах. Трудозатраты с использованием метода (E) раскладываются на составляющие: трудозатраты на проектирование (e_d), накладные затраты на проектирование, трудозатраты на программирование (e_c), накладные затраты на программирование. Каждая из составляющих оценивается в сравнении с трудозатратами на разработку без использования метода (E' , e'_d , e'_c).

Использование разработчиком предложенного в работе метода позволяет фиксировать полученные проектные решения для многоагентных программных систем (роли агентов, модель взаимодействия агентов, интерфейс сервисов) при создании онтологического описания сервисов в унифицированных терминах.

Известно, что использование онтологий на этапе проектирования увеличивает трудозатраты разработчика для создания проектных решений. Однако, одни дополнительные затраты можно минимизировать, а другие дают дополнительные возможности на следующих этапах разработки (например, автоматизация программирования). Одним их способов минимизации затрат является использование существующих инструментальных средств автоматизированного проектирования (например, Protégé). Более того, метод разработки использует предложенные типовые модели взаимодействия агентов и предметно-ориентированные модели сервисов, упрощая проектирование сервисов за счет предоставления разработчику архитектурных и поведенческих абстракций агентов. Таким образом, можно утверждать, что трудозатраты на проектирование сервисов, учитывая накладные затраты, с использованием предложенного метода сопоставимы с трудозатратами на проектирование многоагентных систем без использования метода: $e_d = c * e'_d$, где $1 < c < 2$ – фактор учета накладных затрат.

Дополнительные трудозатраты на проектирование позволяют получать единообразные онтологические модели сервисов, которые определяют интерфейс взаимодействия и описывают семантику построения. Благодаря такому единообразному способу проектирования, сервисы различных предметных областей приобретают возможность взаимодей-

ствовать друг с другом независимо от вычислительной среды, в которой разворачиваются СИИО. Полученные проектные решения в виде онтологий сервисов используются для автоматизации дальнейших процессов программирования сервисов (создание объектной модели и структур данных, методы, функции кода).

Трудозатраты на программирование исследуются на основе отношения общего количества строк исходного кода агентов к автоматически созданному с помощью полученной реализации генератора программного кода. В соответствии с таблицей 2 представлена доля сгенерированного программного кода для агентов, участвующих в реализации сервисов. Сервисы разработаны на основе предложенных предметно-ориентированных моделей. Программный код в независимости от роли программного агента состоит из следующих блоков: (1) структуры объектной модели данных и методы для работы с ними, (2) информационно-управляемое взаимодействие на основе поддерживаемых операций для промежуточного программного обеспечения, (3) внутренняя логика, включающая локальную обработку общей информации. Средняя доля сгенерированного программного кода составила 23,4%, причем наибольшие результаты пришлось на блок «информационно-управляемое взаимодействие».

Таблица 2 – Доля сгенерированного программного кода для реализаций сервисов

Сервис	Агенты и их роли		Объектная модель данных		Информационно-управляемое взаимодействие		Внутренняя логика		Итого	
			SLOC	%	SLOC	%	SLOC	%	SLOC	%
Сервис распознавания присутствия и анализа активности	Агент-адаптер сенсора	общ.	26	8,8	78	26,4	191	64,8	295	100
		сген.	21	7,1	40	13,6	9	3	70	23,7
	Агент-агрегатор присутствия	общ.	18	11,2	44	27,5	98	61,2	160	100
		сген.	14	8,7	18	11,3	5	2,8	37	23,1
	Агент-монитор активности	общ.	74	13,2	88	15,7	392	71,1	560	100
		сген.	52	9,3	75	13,4	13	2,3	140	25
Сервис сопровождения и визуализации плана деятельности	Агент-визуализатор программы СД	общ.	314	14,9	519	24,6	1274	60,5	2107	100
		сген.	135	6,4	280	13,3	27	1,3	442	20,9
	Агент-контроллер управления программой СД	общ.	316	12,8	916	37,1	1236	50,1	2468	100
		сген.	210	8,5	540	21,9	22	0,9	772	31,3
	Агент-интерпретатор веб-сервиса управления содержимым	общ.	92	8,2	217	19,3	817	72,5	1126	100
		сген.	60	5,3	106	9,4	8	0,7	174	15,4
	Агент-воспроизводитель мультимедийной информации	общ.	283	9,3	613	20,1	2153	70,6	3049	100
сген.		186	6,1	253	8,3	37	1,2	476	15,6	
Сервис совместного пополнения содержимого историческими данными	Агент-искатель исторической информации	общ.	65	9,6	162	23,9	451	66,5	678	100
		сген.	50	7,4	103	15,2	7	1	160	23,6
	Агент-контроллер исторических связей	общ.	222	12,6	515	29,2	1028	58,2	1765	100
		сген.	148	8,4	339	19,2	23	1,3	510	28,9
	Агент-агрегатор обогащения информации	общ.	391	11,9	898	27,3	2001	60,8	3290	100
		сген.	296	9	513	15,6	26	0,8	835	25,4

Сервис мониторинга объектов физической среды	Агент-интерпретатор базы данных сенсорной сети	общ.	82	11,6	144	20,3	484	68,1	710	100
		сген.	53	7,5	106	14,9	9	1,3	168	23,7
	Агент-агрегатор измерений сенсорной сети	общ.	82	9,8	194	23,2	562	67	838	100
		сген.	52	6,2	106	12,6	5	0,6	163	19,4
	Агент-контроллер узлов сенсорной сети	общ.	40	12,7	89	28,5	183	58,8	312	100
		сген.	29	9,3	54	17,4	6	1,9	89	28,6

На процесс программирования сервисов с использованием метода накладываются дополнительные трудозатраты, связанные с временем, затраченным на процесс генерации программного кода. Генератор принимает в качестве входных параметров разработанные онтологии сервисов и онтологию предметной области. Время генерации программного кода зависит от онтологических метрик (цикломатическая сложность, размер словаря) характеризующих сложность обхода метамодели онтологии (онтологический граф). Эксперименты показали, что при обработке онтологий реализованных сервисов время обхода соответствующих метамodelей и генерации кода не превышает 12 с.

Качество сгенерированного кода исследуется на основе исследования работы экспериментальных образцов предметно-ориентированных сервисов. В частности, в качестве иллюстрирующего примера рассмотрена организация проведения конференций в интеллектуальном зале. Экспериментальные исследования сервиса сопровождения и визуализации программы деятельности выполнялись на основе оценки возможностей (функций сервиса) по формированию программы деятельности людей, обработке программы по ходу деятельности, показу медиа- и другой визуальной информации.

Одними из ресурсоемких операций для сервиса являются (i) получение программы конференции из информационного содержимого с помощью сетевого взаимодействия (время T_{ntw}) и (ii) локальная обработка программы конференции (время T_{loc}). Экспериментальная оценка $T_{loc}(n)$ представлена в соответствии с рис. 9. Экспериментальная оценка $T_{ntw}(n)$ представлена на рис. 10. Число докладов изменяется в интервале $2 \leq n \leq 50$, чтобы проанализировать масштабируемость для разнообразных форматов секции. Для каждого n производится 100 измерений, в то время как график показывает среднее поведение с помощью стандартного отклонения. Представленные графики описывают линейные и полиномиальные регрессии. Использовалась беспроводная сеть с ограниченной пропускной способностью. Установлено, что для проведения в интеллектуальном зале конференции типичного размера (8-16 докладов) сумма времени локальной обработки и времени передачи не превышает 2 с.

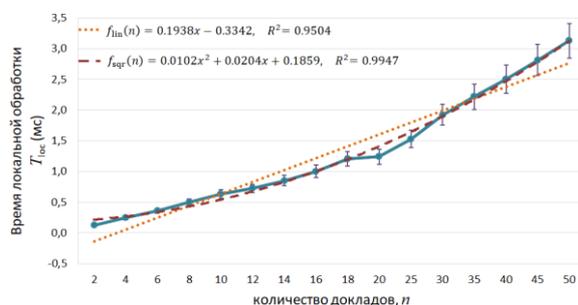


Рисунок 9 – Экспериментальное время T_{loc} локальной обработки программы конференции, состоящей из n докладов

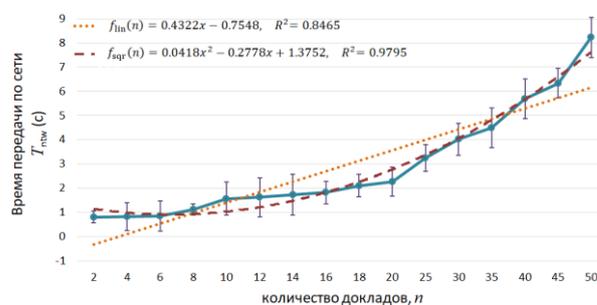


Рисунок 10 – Экспериментальное время T_{ntw} получения программы (n докладов) из информационного содержимого

Таким образом, показано, что использование предложенного метода позволяет снизить трудозатраты на разработку программной инфраструктуры для сервисов СИИО за

счет унифицированной и автоматизированной разработки сервиса как многоагентной системы, интегрирующей динамические и неоднородные ресурсы при построении сервиса.

ЗАКЛЮЧЕНИЕ

В диссертационной работе предложено решение актуальной научно-технической задачи по повышению эффективности разработки интероперабельной программной инфраструктуры совместно используемого информационного интернет-окружения за счет унифицированного моделирования сервиса как системы взаимодействующих агентов и автоматизированного программирования взаимодействия на основе кодогенерации. Получены следующие результаты:

1. Сформирован метод разработки интероперабельной программной инфраструктуры, обеспечивающей информационно-управляемое взаимодействие агентов для интеграции динамических и неоднородных ресурсов при построении сервиса.

2. Предложена концептуальная модель сервиса, позволяющая описывать с помощью онтологии варианты информационно-управляемого взаимодействия агентов для построения контекстных сервисов и их композиции на основе технологий Семантического веба.

3. Разработан алгоритм автоматизации программирования взаимодействия агентов, обеспечивающий кодогенерацию программных механизмов информационно-управляемого взаимодействия по заданной онтологии информационного сервиса.

4. Предложен набор предметно-ориентированных моделей проектирования сервисов СИИО в качестве шаблонных решений для востребованных приложений.

5. Реализован комплекс программных средств для сформированного метода разработки программной инфраструктуры.

В качестве **рекомендаций и перспектив дальнейшей разработки темы** можно указать научное направление исследования самоорганизации агентов путем определения их функциональных ролей, онтологического моделирования семантики взаимодействия агентов и требуемых операций при построении сервиса, а также разработки программных средств оперативного сбора сведений о состоянии и доступных ресурсах.

Полученные результаты соответствуют п. 3 «Модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем» и п. 8 «Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования» паспорта специальности 05.13.11 — «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

.СПИСОК ОСНОВНЫХ ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

В рецензируемых журналах из списка ВАК и изданиях, приравненных к ним:

1. **Марченков С. А.** Автоматизация процессов программирования агентов на основе кодогенерации при построении семантических сервисов интеллектуальных пространств. Часть 1 // Программная инженерия. – 2019. – Т. 10. – №. 6. – С. 257-264. (**ВАК**)
2. **Marchenkov S. A.,** Korzun D. G., Shabaev A. I., Voronin A. V. On applicability of wireless routers to deployment of smart spaces in Internet of Things environments // 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). – IEEE, 2017. – Vol. 2. – P. 1000-1005. (**Web of Science, Scopus**)
3. **Marchenkov S.A.,** Vdovenko A.S., Petrina O.B., Korzun D.G. Smart museum of everyday life history in Petrozavodsk State University: Software design and implementation of the semantic layer // 21st Conference of Open Innovations Association (FRUCT). – IEEE, 2017. – P. 224-230. (**Scopus**)
4. **Марченков С. А.,** Вдовенко А. С., Корзун Д. Ж. Расширение возможностей совместной деятельности в интеллектуальном зале на основе сервисов электронного туризма // Труды СПИИРАН. – 2017. – Т. 1. – №. 50. – С. 165-189. (**ВАК, Scopus**)

5. Korzun D. G., **Marchenkov S. A.**, Vdovenko A. S., Petrina O. B. A semantic approach to designing information services for smart museums // International Journal of Embedded and Real-Time Communication Systems (IJERTCS). – 2016. – Vol. 7. – No. 2. – P. 15-34. (**Scopus**)
6. Korzun D. G., **Marchenkov S. A.**, Vdovenko A. S., Borodulin A. N., Balandin S. I. Performance evaluation of Smart-M3 applications: A SmartRoom case study // 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT). – IEEE, 2016. – P. 138-144. (**Web of Science, Scopus**)
7. **Марченков С. А.**, Корзун Д. Ж. Определение присутствия пользователей в интеллектуальном зале на основе отслеживания активности в беспроводной сети // Ученые записки Петрозаводского государственного университета. Сер.: Физико-математические науки. – 2015. – №. 2. – С. 114-119. (**ВАК**)
8. **Marchenkov S.**, Korzun D. User presence detection based on tracking network activity in smartroom // Proceedings of 16th Conference of Open Innovations Association FRUCT. – IEEE, 2014. – P. 45-50. (**Web of Science, Scopus**)

Зарегистрированные результаты интеллектуальной деятельности:

9. Петрина О. Б., **Марченков С. А.**, Корзун Д. Ж. Свидетельство о государственной регистрации программы для ЭВМ № 2018664673, Российская Федерация. Программный модуль локального ранжирования данных для составления персональных рекомендаций на основе онтологической базы знаний истории повседневности Петрозаводского государственного университета. – Зарег. 20.11.2018.
10. Вдовенко А. С., **Марченков С. А.**, Петрина О. Б., Корзун Д. Ж. Свидетельство о государственной регистрации программы для ЭВМ № 2018610213, Российская Федерация. Мобильный клиент для семантического аннотирования и персонализированного доступа к корпусу источников по истории повседневности в музее (версия для операционной системы Android). – Зарег. 09.01.2018.
11. Петрина О. Б., Волохова В. В., Яловицына С. Э., Каюмова М. Р., Семьина Д. А., **Марченков С. А.**, Вдовенко А. С., Варфоломеев А. Г., Корзун Д. Ж. Свидетельство о государственной регистрации базы данных № 2017621001, Российская Федерация. Онтологическая база знаний истории повседневности Петрозаводского государственного университета. – Зарег. 01.09.2017.
12. Ломов А. А., **Марченков С. А.**, Корзун Д. Ж. Свидетельство о государственной регистрации программы для ЭВМ № 2016662271, Российская Федерация. Инструментальное средство C_KPI платформы Smart-M3 для реализации на языке программирования ANSI C доступа на уровне модели RDF к информационному содержанию интеллектуального пространства. – Зарег. 15.09.2016.
13. **Марченков С. А.**, Петрина О. Б., Ломов А. А., Варфоломеев А. Г., Кулаков К. А. Свидетельство о государственной регистрации программы для ЭВМ № 2016611636, Российская Федерация. Программный модуль поиска и извлечения историко-культурной информации из базы знаний DVpedia для представления в интеллектуальном пространстве. – Зарег. 15.12.2015.
14. **Марченков С. А.**, Вдовенко С. А., Корзун Д. Ж. Свидетельство о государственной регистрации программы для ЭВМ № 2016614670, Российская Федерация. Сервис динамического формирования контекстно-ориентированных веб-страниц в интеллектуальном зале SmartRoom. – Зарег. 13.11.2015.
15. **Марченков С. А.**, Вдовенко С. А., Корзун Д. Ж. Свидетельство о государственной регистрации программы для ЭВМ № 2016610808, Российская Федерация. Сервис историко-культурного сопровождения коллективной работы в интеллектуальном зале SmartRoom. – Зарег. 10.11.2015.

Автореферат диссертации

МАРЧЕНКОВ
Сергей Александрович

АВТОМАТИЗИРОВАННАЯ РАЗРАБОТКА ИНТЕРОПЕРАБЕЛЬНОЙ
ПРОГРАММНОЙ ИНФРАСТРУКТУРЫ ДЛЯ ОРГАНИЗАЦИИ СОВМЕСТНО
ИСПОЛЬЗУЕМОГО ИНФОРМАЦИОННОГО ИНТЕРНЕТ-ОКРУЖЕНИЯ

Текст автореферата размещен на сайтах:

Высшей аттестационной комиссии Министерства образования и науки Российской
Федерации

<https://vak.minobrnauki.gov.ru>

Федерального государственного бюджетного учреждения науки Санкт-
Петербургского института информатики и автоматизации Российской академии наук
(СПИИРАН)

<http://www.spiiras.nw.ru/dissovet/marchenkov-info/>

Подписано в печать «__» _____ 2019 г.
Формат 60x84 1/16. Бумага офсетная. Печать офсетная.
Усл.печ.л. 1,0. Тираж 100 экз.
Заказ № __