

На правах рукописи



Галов Иван Викторович

**Модели проектирования программной инфраструктуры
интеллектуального пространства для
ресурсно-ограниченных вычислительных сред**

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

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

кандидата технических наук

Петрозаводск – 2017

Работа выполнена в Петрозаводском государственном университете.

Научный руководитель: кандидат физико-математических наук,
доцент кафедры информатики и математического
обеспечения ФГБОУ ВО «Петрозаводский государ-
ственный университет» (ПетрГУ)

Корзун Дмитрий Жоржевич

Официальные оппоненты: доктор физико-математических наук, профессор, за-
ведующий кафедрой системного программирования
ФГБОУ ВО «Санкт-Петербургский государственный
университет» (СПбГУ)

Терехов Андрей Николаевич

кандидат технических наук, доцент кафедры вычис-
лительной техники ФГАОУ ВО «Санкт-Петербур-
гский государственный электротехнический универси-
тет «ЛЭТИ» им. В.И.Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)

Пантелеев Михаил Георгиевич

Ведущая организация: **ФГБУН Институт прикладных математиче-
ских исследований Карельского научного цен-
тра Российской академии наук (ИПМИ КарНЦ
РАН, г. Петрозаводск)**

Защита состоится «21» декабря 2017 г. в _____ часов на заседании диссертационного совета
Д 002.199.01 при Федеральном государственном бюджетном учреждении науки Санкт-Петер-
бургском институте информатики и автоматизации Российской академии наук по адресу:
199178, Санкт-Петербург, В.О., 14 линия, д. 39., ауд. 401.

С диссертацией и авторефератом можно ознакомиться на сайте Федерального государствен-
ного бюджетного учреждения науки Санкт-Петербургского института информатики и авто-
матизации Российской академии наук (СПИИРАН): <http://www.spiiras.nw.ru/dissovet/>

Автореферат разослан «_____» _____ 2017 г.

Ученый секретарь

диссертационного совета Д 002.199.01,

кандидат технических наук



Зайцева А. А.

Общая характеристика работы

Актуальность темы. Развитие технологий Интернета вещей (Internet of Things, IoT) привело к появлению нового класса вычислительных окружений — ресурсно-ограниченных IoT-сред, что позволяет реализовать новые цифровые решения для различных областей экономической и социальной деятельности человека. Особенностью таких сред являются использование коммуникационных сетей с ограничениями (низкая скорость передачи данных) и вовлечение в работу периферийных устройств с ограниченными вычислительными ресурсами. Возникает задача создания в такой среде интеллектуального пространства (ИП) как вычислительного окружения, обеспечивающего динамическое множество участников контекстно-зависимыми информационными сервисами. Участники представлены программными агентами на сетевых вычислительных устройствах (СВУ). Агенты взаимодействуют для совместного построения сервисов, используя доступные ресурсы IoT-среды. В работе рассматриваются ИП с косвенным взаимодействием агентов через информационное хранилище с семантическим представлением состояния IoT-среды и операциями сетевого доступа. При косвенном взаимодействии агент изменяет содержимое информационного хранилища, что воспринимается другими агентами для модификации своего поведения. Необходима программная инфраструктура, обеспечивающая организацию косвенного взаимодействия агентов для совместного построения информационных сервисов. Программная инфраструктура ИП разворачивается на имеющихся в IoT-среде СВУ в виде следующих компонент: а) информационное хранилище для организации косвенного взаимодействия; б) программные агенты для построения сервисов; в) системное программное обеспечение (СПО) для запуска и обеспечения работоспособности как самих агентов, так и информационного хранилища.

Организация косвенного взаимодействия является актуальной задачей для создания ИП в ресурсно-ограниченных IoT-средах в силу следующих условий. Во-первых, известные программные реализации информационных хранилищ (например, Hydra, CoCaMAAL, OSGi SIB) ориентированы на специализированные среды, что затрудняет использование доступного аппаратно- сетевого разнообразия и не обеспечивает возможности развертывания программной инфраструктуры с учетом ограничений на ресурсы IoT-среды. Во-вторых, ресурсно-ограниченная IoT-среда содержит слабопроизводительные СВУ, состав участников подвержен изменениям (многократные подключения и отключения различных СВУ), работа беспроводных сетевых соединений нестабильна, что приводит к сбоям при построении сервисов и не обеспечивает возможности непрерывной работы сервиса (например, на беспроводном маршрутизаторе в режиме 24/7). В-третьих, при создании ИП требуется возможность интеграции с уже разработанными сервисами из других ИП без предопределения набора возможных комбинаций сервисов и используемых предметных областей. В указанных условиях применение существующих моделей для организации косвенного взаимодействия агентов (глобальное облако, общие модели классной доски и публикации/подписки) становится затруднительным, поскольку в этих моделях компоненты программной инфраструктуры не предназначены для работы на слабопроизводительных СВУ.

В диссертационной работе предлагаются оригинальные модели для организации косвенного взаимодействия, на основе которых формируется новый метод разработки программной инфраструктуры ИП для указанных условий ресурсно-ограниченных IoT-сред.

Степень разработанности темы. Развитию методологических основ создания интеллектуальных пространств способствовали труды российских и зарубежных ученых: С. И. Баландина, С. Болдырева, В. И. Городецкого, А. М. Кашевника, Ю. Кильяндера, Д. Ж. Корзуна, Д. Кук, Я. Оливера, А. Л. Ронжина, А. В. Смирнова, Р. М. Юсупова, Т. Чинотти, Ю. Хонкколы, А. Д'Элиа и других. Вопросы разработки программных инфраструктур для применения IoT-технологий рассмотрены в работах Ч. Аггарвала, М. А. Аксеновой, В. А. Александрова, Б. Канга, Х. Кима, Д. И. Муромцева, О. Ю. Никифорова, М. Г. Пантелеева, М. Раззака, А. В. Рослякова, А. Н. Терехова, Д. Э. Трибушкова, А. Эспозито и других. Разработка программной инфраструктуры для создания ИП в ресурсно-ограниченной IoT-среде требует новых методов, обеспечивающих участие слабопроизводительных СВУ в построении информационных сервисов, что показано в работах Ф. Бономи, А. В. Дастджерди, Н. К. Гьянга и других.

Целью работы являются обеспечение технической возможности и повышение эффективности построения информационных сервисов в интеллектуальном пространстве в условиях ресурсно-ограниченных IoT-сред за счет разработки моделей проектирования для организации косвенного взаимодействия программных агентов и формирования метода разработки программной инфраструктуры интеллектуального пространства.

Для достижения цели в работе поставлены следующие задачи:

1. Выполнить обзор моделей проектирования для организации косвенного взаимодействия программных агентов и методов разработки программных инфраструктур ИП.
2. Сформировать метод разработки программной инфраструктуры ИП для построения информационных сервисов на основе моделей проектирования для организации косвенного взаимодействия программных агентов в условиях ресурсно-ограниченных IoT-сред.
3. Разработать модели управления сетевым доступом программных агентов и обеспечения устойчивости программной инфраструктуры ИП к сбоям IoT-среды для организации косвенного взаимодействия агентов через информационное хранилище.
4. Разработать онтологическую модель информационных уведомлений для организации косвенного взаимодействия агентов в ИП на основе семантического описания требуемых вариантов взаимодействия при построении сервиса.
5. Провести экспериментальное исследование свойств программных инфраструктур, разрабатываемых на основе предложенных метода разработки и моделей проектирования для организации косвенного взаимодействия программных агентов в ИП.

Объектом исследования является программная инфраструктура, обеспечивающая создание интеллектуального пространства в ресурсно-ограниченной IoT-среде для построения информационных сервисов на основе косвенного взаимодействия программных агентов.

Предметом исследования являются модели проектирования программного обеспечения для организации косвенного взаимодействия агентов в интеллектуальном пространстве.

Научная новизна работы. Разработаны три оригинальные модели проектирования, формирующие в совокупности новый метод разработки программной инфраструктуры ИП.

1. Предложен метод разработки программной инфраструктуры ИП на основе оригинальных моделей проектирования для организации косвенного взаимодействия программных агентов, отличающийся возможностью создания ИП с использованием широкого разнообра-

зия слабопроизводительных СВУ в IoT-среде, при низкой производительности локальных сетевых коммуникаций и без привлечения множества дополнительных СВУ.

2. Предложена концептуальная модель управления сетевым доступом программных агентов к информационному хранилищу на основе модульной системной архитектуры, отличающаяся возможностью настройки информационного хранилища для выполнения на заданном СВУ IoT-среды за счет выбора поддерживаемых операций сетевого доступа и способа их параллельной обработки.

3. Предложена структурная модель обеспечения устойчивости компонентов программной инфраструктуры к сбоям IoT-среды на основе многоуровневой системы восстановления, отличающаяся возможностью использования имеющихся в IoT-среде СВУ без привлечения множества дополнительных СВУ или программных агентов для восстановления на следующих уровнях: а) операции подписки; б) сетевых соединений между агентами и информационным хранилищем; в) вычислительных процессов работы агентов на СВУ; г) хранения данных с разделением семантической информации и объемных бинарных данных.

4. Предложена онтологическая модель информационных уведомлений на основе семантического описания требуемых вариантов взаимодействия для заданного ИП, отличающаяся возможностью унифицированного событийно-ориентированного программирования индивидуального участия агентов во взаимодействии при построении ими информационного сервиса, что позволяет упростить разработку и интеграцию в ИП информационных сервисов.

5. Реализован комплекс программных средств в соответствии с предложенными методом разработки программной инфраструктуры и моделями проектирования программного обеспечения для организации косвенного взаимодействия агентов, в составе: а) реализация информационного хранилища CuteSIB, отличающаяся возможностью создания и настройки ИП для существующего разнообразия ресурсно-ограниченных IoT-сред; б) реализация компонентов программной инфраструктуры на примере системы проведения мероприятий совместной деятельности, отличающаяся возможностями использования слабопроизводительных СВУ для построения опорных сервисов системы, восстановления после сбоев IoT-среды и расширения системы за счет интеграции сервисов из других ИП и сети Интернет; в) реализация ПО для проведения экспериментального исследования по оценке возможностей применения предложенных моделей.

Практическая значимость работы. Предложенный метод позволяет разработать программную инфраструктуру для создания требуемого ИП в заданной ресурсно-ограниченной IoT-среде с целью построения информационных сервисов для различных областей деятельности человека. Реализовано информационное хранилище CuteSIB как основное программное средство для создания ИП в разнообразных ресурсно-ограниченных IoT-средах. В качестве референтного ИП исследована система проведения мероприятий совместной деятельности, реализующая автоматизированную поддержку проведения конференций, совещаний и других мероприятий. Для этого ИП разработан вариант программной инфраструктуры, обеспечивающий построение опорных сервисов системы и демонстрирующий применение предложенных метода и моделей. Проведенное экспериментальное исследование предоставляет оценки эффективности и возможности применения предложенных моделей для организации косвенного взаимодействия агентов в условиях ресурсно-ограниченных IoT-сред.

Методы исследования. Результаты выполненных в работе исследований и программных разработок основаны на методах системного и распределенного программирования, сервисно-ориентированных информационных систем, онтологического моделирования, а также на Интернет-технологиях и технологиях Семантического Веба.

Положения, выносимые на защиту.

1. Метод разработки программной инфраструктуры интеллектуального пространства для построения информационных сервисов взаимодействующими агентами в условиях разнообразия и нестабильности ресурсно-ограниченных IoT-сред.

2. Концептуальная модель управления сетевым доступом программных агентов к информационному хранилищу для организации косвенного взаимодействия агентов на основе выбора поддерживаемых хранилищем операций доступа и настраиваемых способов параллельной обработки информации.

3. Структурная модель обеспечения устойчивости компонентов программной инфраструктуры к сбоям для поддержки взаимодействия агентов на основе многоуровневой системы восстановления после сбоев.

4. Онтологическая модель информационных уведомлений для организации косвенного взаимодействия агентов в ИП на основе семантического описания требуемых вариантов взаимодействия при построении сервиса.

5. Комплекс программных средств, разработанных в соответствии с предложенными методом разработки программной инфраструктуры ИП и моделями проектирования ПО для организации косвенного взаимодействия агентов.

Степень достоверности и апробация. Достоверность научных положений, выводов и результатов диссертационной работы обеспечивается за счет анализа состояния исследований в данной области, согласованности теоретических свойств предложенных моделей проектирования ПО с результатами экспериментального исследования полученной программной реализации, а также апробацией основных положений диссертации в печатных трудах и докладах на научных конференциях. Новизна технических решений подтверждается двумя полученными свидетельствами о регистрации программ для ЭВМ. Результаты работы представлялись на международных конференциях ассоциации открытых инноваций FRUCT в 2010–2016 гг.; международных конференциях ruSMART (2011, 2012, 2014 гг.); международной конференции Information Technology Interfaces в 2012 г.; международных семинарах Annual International Workshop on Advances in Methods of Information and Communication Technology в 2013, 2014, 2015 гг.; всероссийской конференции «Современные технологии в теории и практике программирования» в 2013, 2014, 2015 гг.

Реализация результатов работы. Работа выполнена в рамках федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» – соглашение № 14.574.21.0060 (RFMEFI57414X0060). Исследование поддержано: 1) Международной программой Karelia ENPI – грант КА179, 2012–2014 гг.; 2) Минобрнауки России – НИР № 2336 проектной части государственного задания № 2.2336.2014/К в сфере научной деятельности, 2014–2016 гг.; 3) РФФИ – научный проект № 14-07-00252, 2014–2016 гг.; 4) Минобрнауки России – НИР № 1481 базовой части государственного задания № 2014/154 в сфере научной деятельности,

2014–2016 гг.; 5) Программой стратегического развития ПетрГУ на 2012–2016 гг. Результаты диссертационной работы внедрены в учебный процесс ПетрГУ, применяются в рабочем процессе ООО «Опти-Софт», используются для проведения международных конференций Ассоциацией Открытых Инноваций FRUCT. Предложенные модели и метод применяются в экспериментальных исследованиях по заданию № 2.5124.2017/БЧ Минобрнауки России в рамках базовой части государственного задания на 2017–2019 гг.

Публикации. По материалам диссертации опубликовано 19 печатных работ, включая 3 работы в российских журналах из списка ВАК и 10 работ в международных изданиях, индексируемых в реферативных базах Web of Science или Scopus.

Структура и объем диссертации. Диссертация объемом 150 машинописных страниц состоит из введения, 4 глав, заключения, списка использованной литературы (129 наименований) и 3 приложений. Текст диссертации содержит 39 рисунков и 13 таблиц.

Содержание работы

Во введении обосновывается актуальность диссертационной работы, формулируются цель и задачи исследования, приводятся выносимые на защиту положения, их научная новизна и практическая значимость.

В первой главе вводится исследуемый класс интеллектуальных пространств со следующими свойствами: а) косвенное взаимодействие программных агентов с ограниченной автономностью на основе совместного накопления и обработки информации через информационное хранилище; б) семантическое представление содержимого информационного хранилища для обеспечения его связывания и интероперабельного использования; в) организация взаимодействия на основе набора операций сетевого доступа агентов к информационному хранилищу, включая семантический поиск и постоянные запросы. Применение таких ИП возможно для широкого круга сервисно-ориентированных приложений. При косвенном взаимодействии не требуется установка попарных сетевых соединений между участвующими СВУ, за счет чего обеспечивается снижение потребления ограниченных ресурсов сети и СВУ. Ограниченная автономность агентов необходима для использования СВУ, которые не обладают достаточными вычислительными ресурсами для реализации агентом самостоятельной постановки цели и принятия решений.

Информационное хранилище, развернутое в заданной IoT-среде, определяет общую для агентов точку сетевого доступа к совместно накапливаемой и обрабатываемой информации. Семантическое представление содержимого *I* информационного хранилища реализуется на основе модели RDF в виде базы данных RDF-троек (БД RDF-троек, от англ. Resource Description Framework). Структура *I* описывается онтологиями на языке OWL в терминах классов, свойств и индивидов (экземпляров класса). Хранилище поддерживает такие операции сетевого доступа как добавление (insert), удаление (remove), обновление (update), чтение (query), подписка на изменения (subscription) и другие. Каждый агент индивидуально формирует и отправляет сетевые запросы к информационному хранилищу на выполнение операций. Операция подписки является постоянным запросом и позволяет организовать взаимодействие агентов для проактивной (упреждающей) доставки сервисов.

Программная инфраструктура ИП включает следующие компоненты: а) информацион-

ное хранилище для организации косвенного взаимодействия; б) программные агенты для построения сервисов; в) сервисное системное ПО (СПО) для обеспечения работоспособности хранилища и агентов. Следующие условия затрудняют разработку программной инфраструктуры при создании заданного ИП в ресурсно-ограниченной IoT-среде.

1. Известные программные реализации информационного хранилища (например, Hydra, CoSaMAAL, OSGi SIB) ориентированы на специализированную среду, что затрудняет создание ИП в других средах. Необходима реализация, применимая для разных ресурсно-ограниченных IoT-сред, включая возможность настройки с учетом аппаратно-сетевых возможностей и ограничений участвующих СВУ.

2. Нестабильность сетевых коммуникаций приводит к сбоям при организации взаимодействия агентов в ИП. Число одновременно участвующих в ИП агентов варьируется из-за мобильности подключения и отключения, в том числе из-за частых сбоев беспроводных сетевых соединений. Необходимо обеспечивать устойчивость к сбоям за счет восстановления работоспособности программных агентов и информационного хранилища.

3. Разнообразие способов программирования взаимодействия для известных реализаций информационного хранилища приводит к трудностям при переносе или интеграции сервиса в другое ИП. Необходим унифицированный способ программирования индивидуального участия агентов в совместном построении сервиса, чтобы сократить трудоемкость разработки приложений, в том числе за счет повторного использования программных компонент.

Существующие методы разработки программной инфраструктуры ИП не учитывают указанных условий. В частности, платформы Hydra, A3-TAG и KSpot⁺ хоть и предназначены для разработки сервисно-ориентированных программных инфраструктур на основе единого информационного пространства, но ориентированы только на среды беспроводных сенсорных сетей (WSN) и не обеспечивают устойчивости к сбоям беспроводной среды. Платформы CoSaMAAL и meShup позволяют разрабатывать инфраструктуру для создания среды повсеместных вычислений с обеспечением устойчивости к сбоям, но ориентированы на конкретные аппаратно-сетевые конфигурации вычислительной среды (поддержка СВУ определенных классов). Известной платформой для разработки программных инфраструктур ИП является платформа Smart-M3, в состав которой входят такие реализации информационных хранилищ как RedSIB, PySIB и OSGi SIB. Поддерживается возможность развертывания компонент программных инфраструктур на ряде классов СВУ со слабыми вычислительными ресурсами. В то же время, эти реализации не обеспечивают приемлемую для многих прикладных задач производительность и непрерывную работоспособность сервисов из-за неэффективности управления сетевым доступом к информационному хранилищу и из-за отсутствия механизмов восстановления после сбоев в IoT-среде.

Выполненный обзор позволяет сделать вывод, что одни из существующих решений направлены на разработку программной инфраструктуры для различных вычислительных сред, но без существенного использования имеющихся в среде ресурсов слабопроизводительных СВУ и без учета разнообразия возникающих сбоев. Другие из существующих решений, хоть и обеспечивают достаточный уровень работоспособности в условиях сбоев, но ориентированы на конкретную аппаратно-сетевую конфигурацию среды и требуют привлечения дополнительных ресурсов в виде специализированных СВУ.

Таким образом, получение метода разработки программной инфраструктуры для создания ИП в ресурсно-ограниченной IoT-среде является актуальной задачей, имеющей важное прикладное значение. В диссертационной работе решение этой задачи сводится к разработке новых моделей проектирования программного обеспечения для организации косвенного взаимодействия программных агентов и применению этих моделей в предлагаемом методе.

Во второй главе представлен новый метод разработки программной инфраструктуры ИП для построения информационных сервисов на основе предлагаемых моделей проектирования для организации косвенного взаимодействия программных агентов (рисунок 1). Концептуальная модель M_{mng} управления сетевым доступом программных агентов к информационному хранилищу предназначена для выполнения настройки информационного хранилища под условия заданной IoT-среды. Настройка осуществляется за счет выбора набора операций и способа их параллельной обработки. Структурная модель M_{rec} обеспечения устойчивости компонентов программной инфраструктуры к сбоям предназначена для построения многоуровневой системы восстановления работоспособности информационного хранилища и агентов после сбоев с использованием СПО в IoT-среде. Онтологическая модель M_{ntf} информационных уведомлений предназначена для унификации программирования индивидуального участия агентов в построении информационного сервиса s_i с представлением взаимодействия агентов непосредственно в I . В частности, обеспечивается возможность интеграции сервисов на основе связывания соответствующих информационных областей $I(s_i) \leftrightarrow I(s_j)$.

Метод определяет следующие этапы разработки программной инфраструктуры: а) разработка информационного хранилища (системный программист); б) настройка и запуск информационного хранилища с созданием требуемого ИП в выбранной IoT-среде (администратор ИП); в) описание требуемых вариантов взаимодействия агентов в виде системы инфор-

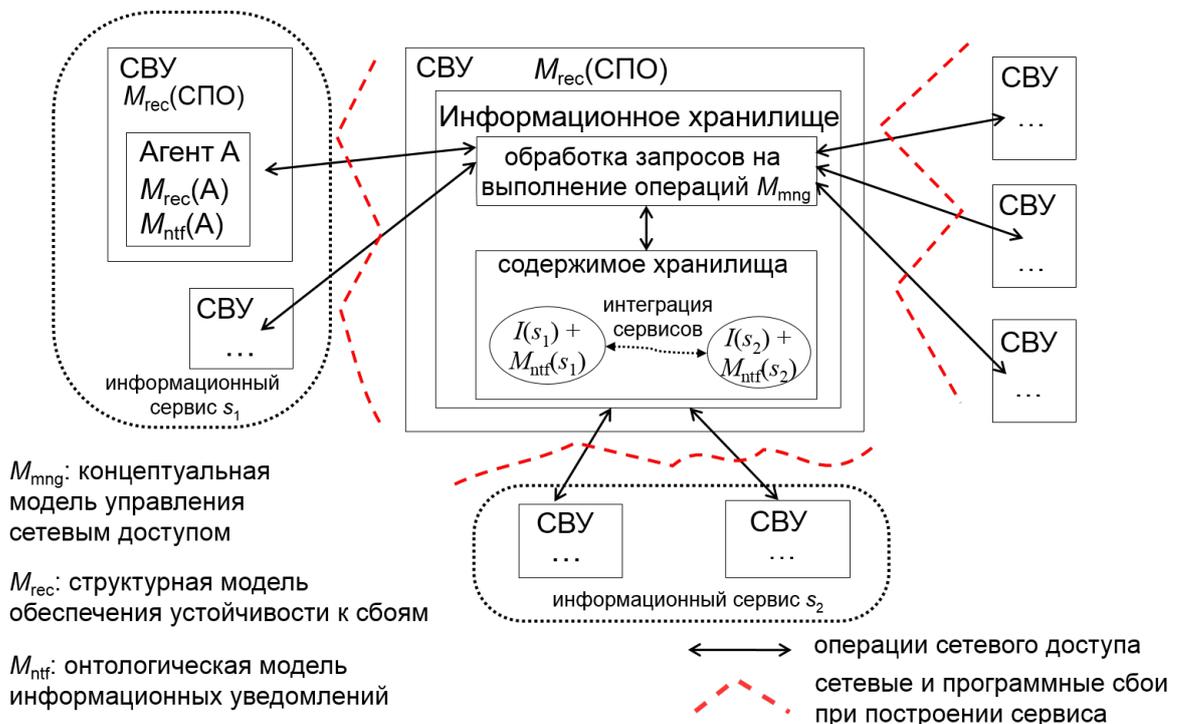


Рисунок 1 – Применение моделей проектирования M_{mng} , M_{rec} и M_{ntf} в методе разработки программной инфраструктуры ИП

мационных уведомлений и реализация взаимодействия для каждого агента в соответствии с этой системой (прикладной программист, администратор ИП); г) реализация программных механизмов обеспечения устойчивости к сбоям и запуск на используемых агентах СВУ (прикладной программист, администратор ИП).

Концептуальная модель управления сетевым доступом программных агентов к информационному хранилищу описывает организацию вычислений в информационном хранилище, требуемую для косвенного взаимодействия агентов и настраиваемую под условия IoT-среды (рисунок 2). Модель включает архитектуру информационного хранилища и способы настройки его параметров. Модель имеет вид $(D, C_D(R))$, где $D = (r, s, o_i, t)$ — последовательность модулей обработки операций для выбираемого набора $i = 1, 2, \dots, n$ операций, C_D — выбираемый способ выполнения вычислений для модулей в D в соответствии с множеством требований $R = (R_{agn}, R_{env}, R_{dnn})$ со стороны агентов, среды и предметной области.

Сетевой доступ и обработка запросов определяются модульной структурой $D = (r, s, o_i, t)$, где r — обработчик сетевых запросов, s — планировщик, o_i — обработчик операции вида i , t — интерфейс взаимодействия с БД RDF-троек. Обработчик запросов принимает сетевые запросы на выполнение операций от множества агентов, преобразует запрос из сетевого формата во внутренний и передает планировщику. Планировщик определяет начало выполнения операции и передает ее на обработку. Для каждой операции i определен собственный обработчик, в соответствии с выбранным набором. Возможны следующие операции: чтение (операция query), изменение (операции добавления insert, удаления remove, обновления update), SPARQL-запросы, подписка, постоянные операции изменения, автоматический вывод зна-



Рисунок 2 – Концептуальная модель управления сетевым доступом программных агентов к информационному хранилищу

ний. Обработчик операций реализует операцию через интерфейс взаимодействия с БД RDF-троек. Последняя реализует низкоуровневые функции добавления и удаления RDF-троек. Результат возвращается через интерфейс взаимодействия с БД RDF-троек и обработчик операций в планировщик, который передает его обработчику запросов для отправки агенту.

Предложенная модель позволяет управлять сетевым доступом агентов за счет настройки информационного хранилища в соответствии с тремя группами требований R : аппаратно-сетевые возможности среды выполнения R_{env} , требования программных агентов R_{agn} , требования предметной области R_{dmn} . Аппаратно-сетевые возможности отражают ограничения на имеющиеся ресурсы среды (например, мощность процессора, объем оперативной памяти), влияющие на возможность запуска и настройку информационного хранилища. Требования программных агентов отражают индивидуально требуемые агентом возможности информационного хранилища (например, поддерживаемые сетевые протоколы, операции доступа). Требования предметной области определяют требования информационного сервиса в целом (например, ограничения на время выполнения). Настройка включает в себя выбор набора поддерживаемых операций и способа их параллельной обработки.

Выбор набора операций сетевого доступа реализуется за счет подключения нужных обработчиков операций. В частности, управление сетевым доступом агентов при помощи расширения набора операций приводит к программной инфраструктуре, обеспечивающей развитые варианты косвенного взаимодействия агентов за счет усиления требований к вычислительным ресурсам СВУ, на котором запущено информационное хранилище.

Управление сетевым доступом агентов при помощи настройки последовательного или параллельного вариантов обработки операций в каждом модуле хранилища позволяет учесть вычислительные возможности выбранного СВУ. Параллельная обработка сетевых запросов, поступающих от множества взаимодействующих агентов, позволяет контролировать отдельные сетевые соединения с агентами, снизить время отправки оповещений при выполнении операции подписки и повысить справедливость обслуживания (примерно равное время ответа). Параллельная обработка операций позволяет повысить производительность обработки операций в случае, если БД RDF-троек поддерживает параллельный доступ.

Структурная модель обеспечения устойчивости компонентов программной инфраструктуры к сбоям описывает возможные уровни для программной инфраструктуры с целью размещения программных механизмов восстановления после сбоев с привязкой к участвующим СВУ и взаимодействующим агентам при построении ими информационных сервисов (рисунок 3). Предлагается выполнять восстановление после сбоев на следующих уровнях: а) уровень операции подписки; б) уровень сетевого соединения; в) уровень вычислительного процесса агента на СВУ и г) уровень хранения данных.

На уровне операции подписки обеспечивается устойчивость к потерям оповещений подписки об изменениях в информационном хранилище. Такие сбои, например, возникают из-за нехватки вычислительных ресурсов на стороне информационного хранилища или из-за нестабильности беспроводных коммуникаций. Обнаружение сбоев выполняется на стороне агента с использованием программного механизма контроля подписки SC. С его помощью агент сам производит проверку на наличие изменений в информационном хранилище (например, через регулярные промежутки времени). Обнаружение агентом изменений является

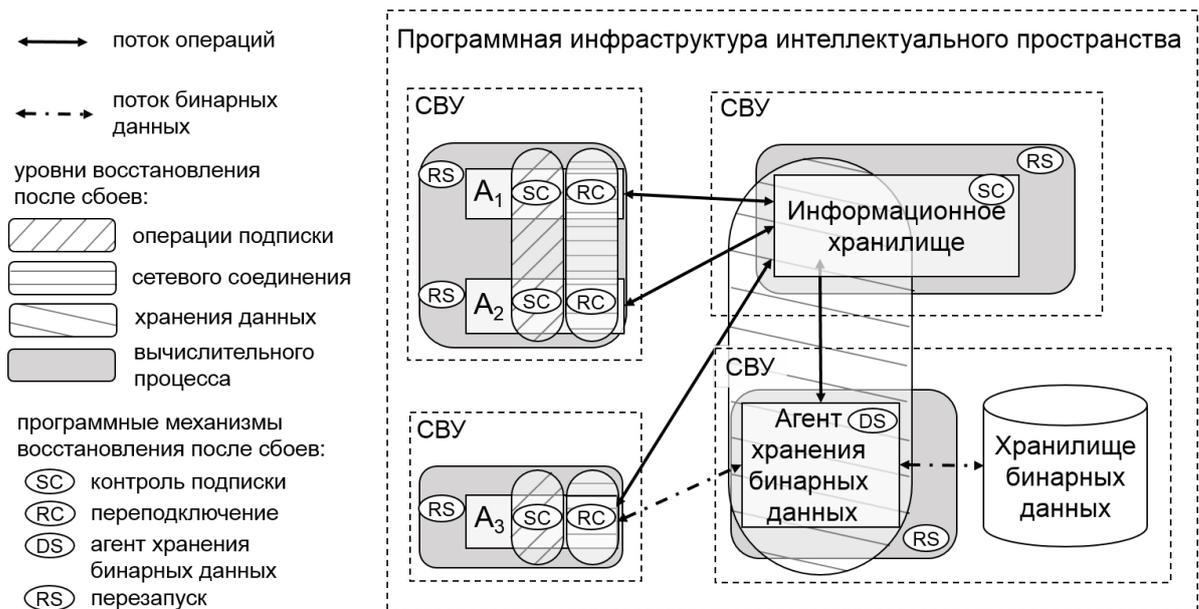


Рисунок 3 – Структурная модель обеспечения устойчивости компонентов программной инфраструктуры интеллектуального пространства к сбоям

и восстановлением, т.к. агент при проверке получает последние внесенные изменения. При этом, часть произошедших изменений может быть потеряна для агента, если было несколько последовательных изменений до момента восстановления.

На уровне сетевого соединения обеспечивается устойчивость к сбоям при потере связи между агентом и информационным хранилищем. Такие сбои возникают из-за нестабильности беспроводных коммуникаций, например, СВУ переключается с одной беспроводной сети на другую или теряет сигнал сети. Обнаружение сбоев выполняется на стороне агента с использованием программного механизма переподрключения RC. При переподрключении устанавливается новое сетевое соединение с информационным хранилищем, и агент может продолжить совместное построение сервиса с другими агентами. После переподрключения необходимо возобновить ранее установленные подписки (агент выполняет новый запрос).

На уровне вычислительного процесса агента обеспечивается устойчивость к сбоям, связанных с аварийным прекращением работы агента. Такие сбои возникают, например, из-за перезагрузки СВУ или временного отказа в ресурсах в силу их необходимости для выполнения основных функций СВУ. Обнаружение сбоев выполняется на стороне СВУ агента с использованием программного механизма перезапуска RS. С его помощью агенты запускаются заново (средствами программной среды СВУ, например, средствами ОС) и восстанавливают предыдущее состояние построения сервиса (до возникновения сбоя). При этом, сетевые соединения с информационным хранилищем устанавливаются повторно.

На уровне хранения данных в программную инфраструктуру вводится агент хранения бинарных данных (изображения, аудио, видео и т.п.). Другие агенты при работе с такими данными могут использовать программный механизм DS для отдельного хранения бинарных данных и их семантики. С его помощью обеспечивается дополнительная защита от сбоев при передаче объемных данных по сети в силу использования специализированного хранилища (например, для мультимедийной информации). За сохранение и организацию доступа отвечает вводимый агент. Он получает сами данные от других агентов и публикует в *I* се-

мантическую информацию в виде ссылок, определяющих способ доступа к этим данным.

В третьей главе предлагается онтологическая модель информационных уведомлений для организации косвенного взаимодействия программных агентов при построении информационных сервисов в заданном ИП. Предложенная модель обеспечивает унифицированный способ программирования индивидуального участия агентов во взаимодействии при построении ими информационного сервиса s . Способ позволяет реализовать реактивное (по запросу) и событийно-ориентированное взаимодействие агентов сервиса. Унификация программирования позволяет реализовать интеграцию сервисов из других ИП, а также снизить число подписок до одной для участия отдельного агента во взаимодействии.

Представим уведомление как информационное сообщение, отправляемое агентом A_{snd} при выполнении определенного условия и получаемое агентом A_{rcv} . Каждое уведомление соответствует отдельному варианту взаимодействия N_k между агентами. Предлагаемая модель определяет: а) способ описания (представления) онтологии уведомлений $O_N(s)$ для сервиса s в виде онтологических классов и свойств для организации взаимодействия между двумя или более агентами на основе операции подписки; б) схему использования этой онтологии для программирования взаимодействия в агентах. В модели выделены два типа взаимодействия: а) запрос: агент-получатель A_{rcv} осуществляет требуемые действия, используя исходные данные от агента-отправителя A_{snd} ; б) событие: получатель A_{rcv} реагирует на определенное событие (информацию), о котором информирует отправитель A_{snd} .

Онтологическая модель информационных уведомлений имеет вид $(\sigma(N_k), O_N, Q(N_k))$, где σ — условие, определяющее инициацию взаимодействия N_k агентом A_{snd} , O_N — онтология системы уведомлений, $Q(N_k)$ — реакция агента A_{rcv} на взаимодействие N_k . Онтология системы уведомлений $O_N(s) = \langle C_N, A_N, D_N, R_N \rangle$, где $C_N = \{c_1, \dots, c_M, c_{\text{ex}}\} \cup C'_s$; c_1, \dots, c_M — множество классов системы уведомлений, каждый из которых соответствует отдельному варианту взаимодействия, c_{ex} — дополнительный класс, используемый для хранения параметров уведомления, $C'_s \subset C_s$ — подмножество классов из онтологии сервиса s , используемых для передачи информации в параметрах уведомления при взаимодействии, A_N — множество атрибутов, используемых для передачи текстовых параметров уведомления, D_N — множество доменов — областей допустимых значений атрибутов, R_N — множество отношений (ограничений), которыми связаны классы.

Онтология $O_N(s)$ используется для проблемно-ориентированного проектирования и программирования участия каждого отдельного агента в совместном построении s . При проектировании сервиса определяется система требуемых вариантов взаимодействия агентов N_k . В онтологии уведомлений $O_N(s)$ каждый вариант взаимодействия N_k представлен отдельным классом из C_N . Собственно уведомление представлено в содержимом I в виде индивида соответствующего класса. Такой индивид семантически описывает отдельный акт взаимодействия между A_{snd} и A_{rcv} . Параметры уведомления p_1, p_2, \dots, p_n представлены в I с помощью свойств онтологического класса из C_N . Свойства данных содержат текстовые значения параметров из A_N . Объектные свойства связывают класс уведомления из C_N с классами из C'_s онтологии предметной области сервиса $O(s)$. Таким образом, формируется система информационных уведомлений, которую используют агенты во время построения сервиса.

Предложенная онтологическая модель приводит к следующей схеме (рисунок 4) исполь-

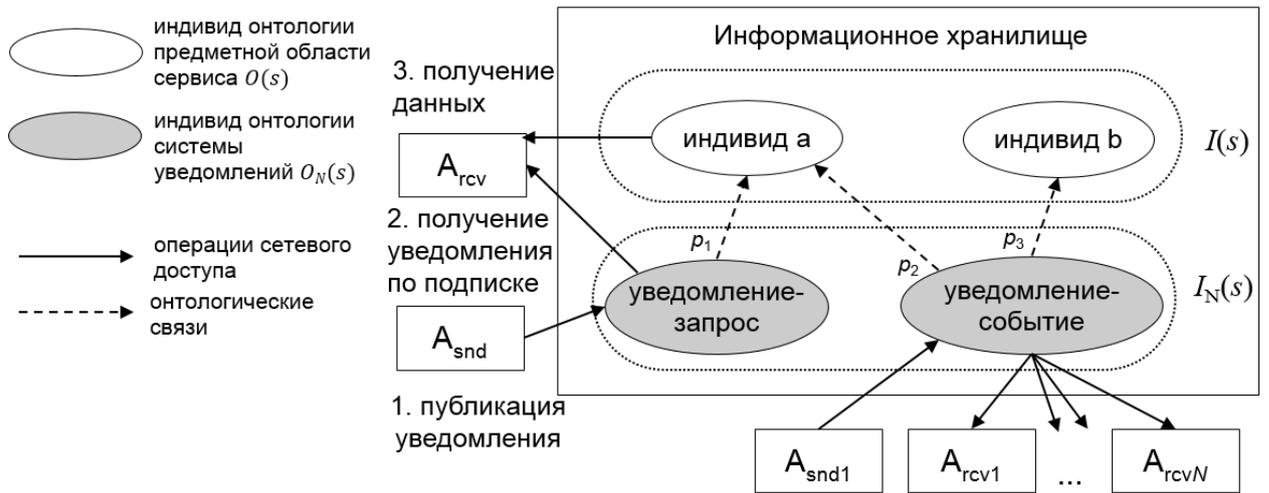


Рисунок 4 – Схема использования уведомлений агентами A_{snd} и A_{rcv}

зования уведомлений для организации взаимодействия агентов. Перед выполнением шагов 1–3 агенты должны подписаться на требуемые им уведомления в $I_N(s)$.

1. Для активации агента A_{rcv} на выполнение действий агент A_{snd} формирует уведомление с n параметрами и публикует индивида уведомления в I . Для передачи дополнительной информации, необходимой в таком варианте взаимодействия $A_{snd} \rightarrow A_{rcv}$, индивид уведомления связывается через свойство-параметр с индивидами предметной области сервиса $I(s)$.

2. Агент A_{rcv} обнаруживает факт публикации уведомления по подписке и получает индивида уведомления.

3. Агент A_{rcv} обрабатывает уведомление: получает значения параметров уведомления; выполняет действия, требуемые данным вариантом взаимодействия, используя исходные данные из полученного уведомления. Затем A_{rcv} удаляет индивида уведомления, если взаимодействие реактивное (по запросу).

Следующие шаги 4–5 выполняются, если необходимо реализовать вариант взаимодействия $A_{snd} \leftrightarrow A_{rcv}$ с обратной связью.

4. Агент A_{rcv} формирует и публикует индивида ответного уведомления с результатами выполнения своих действий.

5. Агент A_{snd} получает индивида ответного уведомления по подписке, реагирует (ответные действия для этого варианта взаимодействия) и удаляет индивида уведомления.

Четвертая глава представляет результаты выполненного экспериментального исследования свойств программных инфраструктур, разрабатываемых на основе предложенного метода разработки и применяемых в нем моделей проектирования для организации косвенного взаимодействия программных агентов. Используется реализация информационного хранилища CuteSIB, полученная при непосредственном участии автора и вошедшая в состав международной технологической платформы Smart-M3. Практическая эффективность также исследуется на примере разработки программной инфраструктуры для создания референтного ИП — системы проведения мероприятий совместной деятельности.

Для исследования производительности операций в модели управления сетевым доступом агентов к информационному хранилищу измерялось время выполнения T_o операций $o \in \{\text{insert, remove, query}\}$ для трех типов СВУ: виртуальная машина (ВМ) на серверной ЭВМ (T_o^{srv}), персональная ЭВМ на примере ноутбука (T_o^{lpt}) и слабопроизводительная (встро-

енная) ЭВМ на примере одноплатного компьютера Raspberry Pi (T_o^{emb}). В соответствии с рисунком 5 (а) можно сделать вывод, что T_{query} не превышает 6 с при работе с числом троек $10^3 \leq N_{\text{RDF}} \leq 35 \cdot 10^3$ на сервере и ноутбуке. Аналогичные результаты получены для T_{insert} и T_{remove} . Встроенное СВУ позволяет работать с объемами до $25 \cdot 10^3$ троек за время, не превышающее 7 с. Показано, что такая производительность приемлема в системах проведения мероприятий совместной деятельности.

Для исследования возможностей настройки параллельной обработки операций в модели управления сетевым доступом агентов измерялось время выполнения операции query для последовательной ($T_{\text{query}}^{\text{seq}}$) и параллельной ($T_{\text{query}}^{\text{prl}}$) обработки. Одновременно запускалось $n = 10$ агентов, каждый из которых осуществлял запрос на $N = 5 \cdot 10^3$ троек. Изменялось время получения этого запроса для каждого из агентов. В соответствии с рисунком 5 (б) подтверждается, что при последовательной обработке для каждого следующего агента время обработки операции линейно возрастает со значимой дифференциацией агентов по времени обслуживания на СВУ с ограниченными ресурсами. При параллельной обработке запросов дифференциация агентов по времени обслуживания проявляется меньше, а общая производительность снижается из-за дополнительных затрат на обеспечение параллельных вычислений на СВУ с ограниченными ресурсами. Наблюдаемый скачок во времени параллельной обработки агентов может быть связан с распределением ресурсов между параллельными процессами. Следовательно, возможности параллельной обработки поддерживают справедливость при сетевом доступе агентов к информационному хранилищу на СВУ с ограниченными ресурсами.

Для исследования повышения производительности за счет использования модели управления сетевым доступом агентов проведено сравнение показателей T_o с другой реализацией информационного хранилища (RedSIB). Изменялось T_o в зависимости от числа RDF-троек N_{RDF} , хранимых в I . При $N_{\text{RDF}} < 10^3$ установлено, что $T_{\text{insert}}^{\text{RedSIB}} \approx T_{\text{insert}}^{\text{CuteSIB}}$. Начиная с $N_{\text{RDF}} > 5 \cdot 10^3$ наблюдается существенное (более чем в 2 раза) превышение $T_{\text{insert}}^{\text{RedSIB}} \gg T_{\text{insert}}^{\text{CuteSIB}}$. Проведено исследование на наличие утечек памяти в предложенной реализации информационного хранилища с измерением объема M_{start} и M_{end} занимаемой хранилищем оперативной памяти до и после периода активной работы с хранилищем. В течение 12 часов с периодичностью 1 раз в минуту добавлялось $N = 5 \cdot 10^4$ троек, которые затем удалялись. Измерения

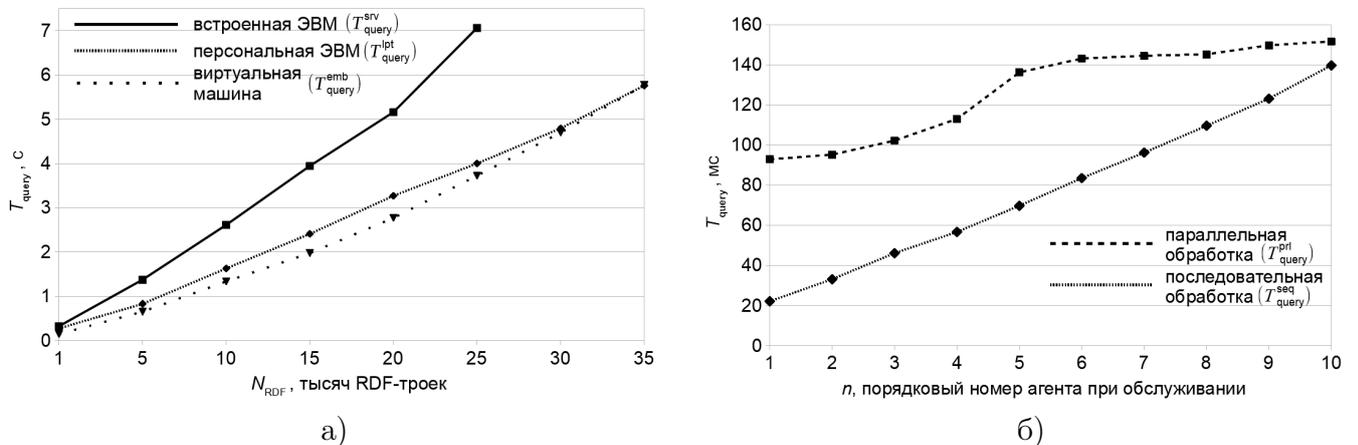


Рисунок 5 – Время выполнения операции query: а) на различных СВУ для запуска информационного хранилища, б) при параллельной и последовательной обработке (на СВУ типа «ноутбук»)

показали, что $M_{\text{start}}^{\text{CuteSIB}} \approx M_{\text{end}}^{\text{CuteSIB}}$, в отличие от реализации RedSIB. В целом, предложенная модель управления сетевым доступом позволяет повысить эффективность построения сервиса за счет увеличения производительности при выполнении операций на СВУ с ограниченными ресурсами, поддержки справедливости доступа агентов к информационному хранилищу и обеспечения долговременной работоспособности информационного хранилища.

Для исследования производительности восстановления после сбоев в модели обеспечения устойчивости программной инфраструктуры измерялось время восстановления работоспособности отдельного программного агента. При перезапуске измерялось время T_{rst} между возникшим сбоем и восстановлением работы агента. При переподключении измерялось время T_{rct} с момента разрыва сетевого соединения агента с информационным хранилищем до полного восстановления соединения. Для агента хранения бинарных данных измерялось время T_{bds} загрузки и сохранения файла размером 10 Мб. В результате получено, что $T_{\text{rst}} \approx 2$ с, $T_{\text{rct}} \approx 0,2$ с, $T_{\text{bds}} \approx 1,8$ с. Показано, что такая производительность приемлема для работы мобильных клиентов в системах проведения мероприятий совместной деятельности, где каждый клиент представляет человека-участника. В целом, предложенная модель обеспечения устойчивости программной инфраструктуры предоставляет техническую возможность создания ИП для долговременной работы в условиях нестабильности IoT-среды.

Применение модели информационных уведомлений упрощает создание ИП за счет применения унифицированного способа программирования участия каждого агента в построении сервиса. Этот эффект подтверждается на примере разработки системы проведения мероприятий совместной деятельности, в которой реализована возможность интеграции существующих сервисов из других предметных областей. Для исследования ухудшения производительности выполнения взаимодействия программных агентов (из-за унификации), измерялось время $T_{\text{ntf}}(n)$ получения агентом уведомления с n параметрами. Поведение $T_{\text{ntf}}(n)$ приближенно соответствует линейной регрессии, где с ростом n увеличиваются время обработки данных на стороне информационного хранилища и объем передаваемых по сети данных. В целом, предложенная модель информационных уведомлений сохраняет приемлемую для референтного ИП производительность. Типичное ограничение $n < 10$ приводит к $T_{\text{ntf}}(n) < 1$ с. Для значений n порядка нескольких десятков (сложность варианта взаимодействия) время $T_{\text{ntf}}(n)$ не превышает нескольких секунд.

Для исследования производительности в целом программной инфраструктуры, разрабатываемой по предложенному методу, проводились имитационные эксперименты на основе разработанного комплекса программных средств. Исследовалась зависимость от числа взаимодействующих с информационным хранилищем агентов трех типов (характерны для многих ИП): 1) сенсоры — обновляют конкретное значение в хранилище с частотой $\lambda_{\text{sns}} \text{ с}^{-1}$ (основная операция сетевого доступа— update); 2) обработчики данных — каждый подписывается на часть информации, регулярно поступающей от сенсоров, и выполняет их обработку с публикацией агрегированного значения (основная операция сетевого доступа— subscribe); 3) клиенты — ожидают агрегированных значений для доставки сервиса пользователю (основная операция сетевого доступа — query). Запускалось n_{sns} агентов-сенсоров, n_{rsn} агентов-обработчиков и n_{cln} агентов-клиентов в пропорции $10^3 : 1 : 10^2$, т.е. 1 обработчик на 10^3 сенсоров и 10^2 клиентов. При этом число обработчиков доходило до нескольких десятков. Такая пропорция

Таблица 1 – Среднее время (в секундах) выполнения операции агентом при интенсивности λ_{sns}

Типы агентов	Число агентов $n_{\text{sns}}, n_{\text{tsn}}, n_{\text{cln}}$ (интенсивность λ_{sns})			
	$10^4, 10, 10^3$ (2 c^{-1})	$5 \cdot 10^4, 50, 5 \cdot 10^3$ (2 c^{-1})	$10^5, 10^2, 10^4$ (2 c^{-1})	$5 \cdot 10^4, 50, 5 \cdot 10^3$ (5 c^{-1})
Сенсоры	0,017	0,034	0,070	0,129
Обработчики	0,059	0,072	0,129	0,101
Клиенты	0,078	0,158	0,265	0,187

отражает предположение: а) обработчик соответствует агенту, ответственному за построение основной части сервиса, б) для построения сервиса нужны данные от множества сенсоров, в) сервис необходим для нескольких клиентов. Измерялось время выполнения основной выполняемой операции для каждого типа агентов (таблица 1). Варьируется число агентов в рамках выбранной пропорции и используются различные значения λ_{sns} . В целом, возможности программной инфраструктуры достаточны для организации взаимодействия агентов в пределах до $10^4 \dots 10^5$ одновременно участвующих СВУ с ограниченными ресурсами.

Полученные выводы по результатам экспериментального исследования сведены в таблицу 2. Применение предложенного метода разработки позволяет обеспечить построение информационных сервисов для рассматриваемых условий ресурсно-ограниченных IoT-сред и с выполнением вычислений на имеющихся СВУ. Обеспечивается возможность запуска информационного хранилища на таких устройствах, как одноплатные компьютеры и сетевые маршрутизаторы. Обеспечивается возможность восстановления компонентов программной инфраструктуры после разнообразных сбоев. В ряде случаев время восстановления может занимать до нескольких секунд, что достаточно для работы мобильных клиентов в системах проведения мероприятий совместной деятельности. Повышается эффективность программной разработки сервисов за счет применения унифицированного способа программирования участия агента во взаимодействии, а производительность построения сервисов при таком взаимодействии повышается за счет снижения числа подписок до одной для отдельного агента.

Заключение

В диссертации предложено решение актуальной задачи по организации косвенного взаимодействия программных агентов интеллектуальных пространств, которое в составе следующих результатов предоставляет возможность построения информационных сервисов в условиях ресурсно-ограниченных IoT-сред.

1. Предложен метод разработки программной инфраструктуры ИП, позволяющий настроить информационное хранилище в соответствии с аппаратно-сетевыми ограничениями IoT-среды и требованиями предметной области, автоматизировать восстановление компонентов программной инфраструктуры после возникающих сбоев, организовать взаимодействие программных агентов при построении сервисов унифицированным способом.

2. Предложена концептуальная модель управления сетевым доступом программных агентов к информационному хранилищу, позволяющая настраивать набор поддерживаемых хранилищем операций сетевого доступа и способов их параллельной обработки для эффективного использования имеющихся вычислительных ресурсов.

Таблица 2 – Результаты экспериментов

Модель	Вид экспериментов	Результат
Управление сетевым доступом	Время выполнения операций T_o , $o = \text{insert, remove, query}$.	- На встроенном СВУ возможна работа при $N_{\text{RDF}} \leq 25 \cdot 10^3$. - Время T_{insert} и T_{remove} увеличивается с ростом числа подписок. - На слабопроизводительных СВУ параллельный вариант обработки повышает справедливость обслуживания агентов на 50% и более. - В сравнении с реализацией RedSIB при $N_{\text{RDF}} > 5 \cdot 10^3$ производительность обработки операций возрастает в 2 и более раз: $T_o^{\text{RedSIB}} \gg T_o^{\text{CuteSIB}}$.
	Наличие утечек памяти.	Обеспечивается непрерывная работоспособность реализации CuteSIB за счет отсутствия утечек памяти: $M_{\text{start}} \approx M_{\text{end}}$.
	Возможности программной инфраструктуры в зависимости от числа агентов.	Время выполнения операций T_o приемлемо (менее 1 с) для референтного ИП при взаимодействии до $10^4 \dots 10^5$ агентов.
Обеспечение устойчивости к сбоям	Время восстановления после сбоев с помощью программных механизмов пересоединения T_{rct} и перезапуска T_{rst} .	Время восстановления после сбоя приемлемо для референтного ИП системы проведения мероприятий совместной деятельности: T_{rct} и T_{rst} не превышают 2 с.
	Производительность агента хранения бинарных данных.	Время загрузки бинарных данных приемлемо для референтного ИП системы проведения мероприятий совместной деятельности: T_{bds} не превышает 2 с для файлов объемом до 10 Мб.
Организация взаимодействия	Время получения уведомления $T_{\text{ntf}}(n)$ в зависимости от количества параметров уведомления n .	- Время обработки уведомлений не приводит к существенным накладным расходам: $T_{\text{ntf}}(n) < 1$ с при $n \leq 10$. - Время выполнения операций T_o уменьшается за счет снижения числа используемых подписок до одной на агента в соответствии с моделью информационных уведомлений.

3. Предложена структурная модель обеспечения устойчивости компонентов программной инфраструктуры к сбоям IoT-среды, позволяющая восстанавливать работоспособность компонентов инфраструктуры на уровнях: а) операции подписки, б) сетевых соединений между агентами и информационным хранилищем, в) вычислительных процессов агентов на СВУ, г) хранения данных.

4. Предложена онтологическая модель информационных уведомлений, позволяющая повысить эффективность процесса программной разработки приложения и упрощающая интеграцию ранее разработанных сервисов.

5. Создан комплекс программных средств в составе: а) реализации информационного хранилища CuteSIB для создания ИП в ресурсно-ограниченных IoT-средах; б) реализации компонентов программной инфраструктуры на примере системы проведения мероприятий совместной деятельности, обеспечивающих построение опорных сервисов с использованием устройств с ограниченными вычислительными ресурсами, устойчивость к сбоям и возможность интеграции сервисов из других ИП; в) реализации ПО для проведения экспериментального исследования по оценке возможностей применения предложенных моделей.

Перспективными направлениями развития полученных научных результатов являются исследование и разработка моделей проектирования для обеспечения информационной безопасности взаимодействия агентов на уровне программной инфраструктуры, а также исследование задачи автоматизации процесса разработки, развертывания и настройки программной инфраструктуры.

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

1. M3-Weather: A Smart-M3 World-Weather Application for Mobile Users / A. Samoryadova, **I. Galov**, P. Borovinskiy et al. // Proc. 8th Conf. of Open Innovations Framework Program FRUCT / Ed. by S. Balandin, Andrei Ovchinnikov. — SUAI, 2010. — Pp. 160–166.
2. Integration of Smart-M3 Applications: Blogging in Smart Conference / D. G. Korzun, **I. V. Galov**, A. M. Kashevnik et al. // Proc. 4th Conf. Smart Spaces (ruSMART'11) and 11th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN'11). — Springer-Verlag, 2011. — Pp. 51–62. (**Web of Science, Scopus**).
3. **Galov I.**, Korzun D. Compositions of Personal Smart Spaces in Multi-Blogging // Proc. 10th Conf. of Open Innovations Association FRUCT and 2nd Finnish–Russian Mobile Linux Summit / Ed. by S. Balandin, Andrei Ovchinnikov. — SUAI, 2011. — Pp. 54–58.
4. Korolev Y., Korzun D., **Galov I.** Smart Space Applications Integration: A Mediation Formalism and Design for Smart-M3 // Proc. 12th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN 2012) & 5th Conf. Internet of Things and Smart Spaces (ruSMART 2012). — LNCS 7469. — Springer-Verlag, 2012. — Pp. 128–139. (**Web of Science, Scopus**).
5. Korzun D. G., **Galov I. V.**, Balandin S. I. Proactive Personalized Mobile Mutli-Blogging Service on Smart-M3 // *Journal of Computing and Information Technology*. — 2012. — Vol. 20, no. 3. — Pp. 175–182. (**Web of Science, Scopus**).
6. Event Recording in Smart Room / **I. Galov**, R. Kadirov, A. Vasilev, D. Korzun // Proc. 13th Conf. of Open Innovations Association FRUCT and 2nd Seminar on e-Tourism for Karelia and Oulu Region / Ed. by S. Balandin, Ulia Trifonova. — SUAI, 2013. — Pp. 20–28.
7. Korzun D., **Galov I.**, Balandin S. Development of smart room services on top of Smart-M3 // Proc. 14th Conf. of Open Innovations Association FRUCT / Ed. by S. Balandin, Ulia Trifonova. — SUAI, 2013. — Pp. 37–44. (**Web of Science, Scopus**).
8. **Galov I.**, Korzun D. The SmartRoom Infrastructure: Service Runtime Reliability // Proc. 14th Conf. of Open Innovations Association FRUCT / Ed. by S. Balandin, Ulia Trifonova. — SUAI, 2013. — Pp. 188–189.
9. Virtual Shared Workspace for Smart Spaces and M3-based Case Study / D. Korzun, **I. Galov**, A. Kashevnik, S. Balandin // Proc. 15th Conf. of Open Innovations Association FRUCT / Ed. by S. Balandin, Ulia Trifonova. — IEEE, 2014. — Pp. 60–68. (**Web of Science, Scopus**).
10. **Galov I.**, Korzun D. A Notification Model for Smart-M3 Applications // Proc. 14th Int'l Conf. Next Generation Wired/Wireless Networking and 7th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2014), LNCS 8638 / Ed. by S. Balandin, S. Andreev, Yevgeni Koucheryavy. — Springer-Verlag, 2014. — Pp. 121–132. (**Scopus**).
11. **Galov I.**, Korzun D. Fault Tolerance Support of Smart-M3 Application on the Software

- Infrastructure Level // Proc. 16th Conf. of Open Innovations Association FRUCT. — 2014. — Pp. 16–23. (**Web of Science, Scopus**).
12. **Галов И.В.**, Корзун Д.Ж. Модель уведомлений для разработки программных приложений интеллектуальных пространств // *Труды СПИИРАН*. — 2014. — Т. 35. — С. 189–211. (**ВАК**).
 13. **Галов И. В.**, Корзун Д. Ж. Обеспечение устойчивости к сбоям Smart-M3 приложения на уровне программной инфраструктуры // *Труды СПИИРАН*. — 2014. — Т. 37. — С. 188—207. (**ВАК**).
 14. **Galov I.**, Lomov A., Korzun D. Design of Semantic Information Broker for Localized Computing Environments in the Internet of Things // Proc. 17th Conf. of Open Innovations Association FRUCT. — IEEE, 2015. — Pp. 36–43. (**Web of Science, Scopus**).
 15. **Галов И. В.** Применение шаблонов проектирования программных приложений для реализации косвенного взаимодействия агентов в интеллектуальном пространстве // *Программная инженерия*. — 2016. — № 8. — С. 351—359. (**ВАК**).
 16. Korzun D. G., **Galov I. V.**, Lomov A. A. Smart Space Deployment in Wireless and Mobile Settings of the Internet of Things // *IDAACS 2016*. — 2016. — P. 86–91. (**Scopus**).
 17. Viola F., D’Elia A., Korzun D., **Galov I.**, Kashevnik A., Balandin S. The M3 Architecture for Smart Spaces: Overview of Semantic Information Broker Implementations // *Proc. 19th Conf. of Open Innovations Association FRUCT*. — 2016. — Pp. 264–272. (**Scopus**).
 18. **Галов И. В.**, Марченков С. А., Корзун Д. Ж. Свидетельство о государственной регистрации программы для ЭВМ № 2015615091, РФ. Программный комплекс опорных сервисов управления программой и материалами докладчиков для проведения мероприятий коллаборативной деятельности типа «конференция» в интеллектуальном зале SmartRoom в составе: Conference-service и Content-service. — Зарег. 07.05.2015.
 19. **Галов И. В.**, Ломов А. А., Корзун Д. Ж. Свидетельство о государственной регистрации программы для ЭВМ № 2017615801, РФ. Программа, реализующая информационное хранилище CuteSIB, для организации косвенного взаимодействия агентов интеллектуального пространства в ресурсно-ограниченных вычислительных средах. — Зарег. 24.05.2017.