

Федеральное государственное автономное образовательное  
учреждение высшего образования  
Санкт-Петербургский национальный исследовательский университет  
информационных технологий механики и оптики

На правах рукописи

**Тесля Николай Николаевич**

**РАЗРАБОТКА МЕТОДОВ И МОДЕЛЕЙ ПОСТРОЕНИЯ  
СЕРВИС-ОРИЕНТИРОВАННОЙ СИСТЕМЫ ОБЕСПЕЧЕНИЯ  
ИНФОМОБИЛЬНОСТИ**

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

Диссертация на соискание ученой степени  
Кандидата технических наук

Научный руководитель  
д.т.н, профессор  
Смирнов Александр Викторович

Санкт-Петербург – 2015

## Оглавление

Введение.....	4
Глава 1. Инфомобильность. Требования к построению системы обеспечения инфомобильности.....	9
1.1 Интеллектуальная транспортная система. Понятие инфомобильности.....	9
1.2 Обзор исследований и проектов в области обеспечения инфомобильности.....	15
1.3 Требования к построению системы обеспечения инфомобильности.....	27
1.4 Технологии, применяемые в проектировании и разработке систем обеспечения инфомобильности.....	33
1.5 Выводы по главе 1 .....	43
Глава 2. Подход и модели построения сервис-ориентированных систем обеспечения инфомобильности.....	45
2.1 Подход к построению систем обеспечения инфомобильности.....	45
2.2 Сценарная модель сервис-ориентированной системы обеспечения инфомобильности .....	50
2.3 Концептуальная модель сервис-ориентированной системы обеспечения инфомобильности .....	55
2.4 Онтологическая модель контекста для сервис-ориентированной системы обеспечения инфомобильности.....	62
2.5 Выводы по главе 2 .....	72
Глава 3. Методы и модели сервисов для системы обеспечения инфомобильности .....	74
3.1 Архитектура системы обеспечения инфомобильности.....	74
3.2 Автоматизированный метод сопоставления онтологий с использованием технологии краудсорсинга .....	81
3.3 Метод планирования мультимодальных маршрутов .....	90
3.4 Метод построения сервиса рекомендаций объектов.....	101
3.5 Модель построения сервиса обеспечения конфиденциальности информации пользователей.....	107
3.6 Выводы по главе 3 .....	114
Глава 4. Реализация сервис-ориентированной системы обеспечения инфомобильности.....	117
4.1 Реализация сервиса планирования маршрутов .....	117
4.2 Реализация ГИС для сервисов планирования маршрута и поиска объектов .....	127
4.3 Реализация сервиса формирования рекомендаций.....	136

4.4	Реализация сервиса сопоставления онтологий .....	142
4.5	Реализация сервиса обеспечения конфиденциальности информации о пользователе .....	150
4.6	Апробация сервис-ориентированной системы обеспечения инфомобильности в области туризма .....	156
4.7	Выводы по главе 4 .....	159
	Заключение .....	162
	Определения .....	164
	Список обозначений и сокращений.....	165
	Литература .....	166
	Приложение 1. Акты внедрения .....	181

## Введение

**Актуальность темы.** В процессе урбанизации многие города столкнулись с проблемой обеспечения удобного и быстрого перемещения населения. Научным сообществом было предложено множество решений данной проблемы, среди которых особое место занимает разработка информационных транспортных сервисов (ИТС): от простых навигационных сервисов до автоматизированных систем управления дорожным движением.

К ИТС также относятся системы обеспечения «инфомобильности» (далее — СОИМ), предоставляющие пользователям доступ с помощью мобильных устройств к мультимодальной динамической персонализированной информации и транспортным сервисам с учетом контекста с целью повышения их мобильности и удобства перемещения. Под контекстом здесь и далее понимается любая информация, которая может быть использована для описания ситуации, в которой находится в определенный момент некоторый объект (человек, транспортное средство или сервис).

СОИМ осуществляют информационную поддержку пользователя как при планировании поездки, так и во время нее, для чего используется широкий набор сервисов, начиная с поиска транспортных средств для совершения поездки и объектов интереса на пути пользователя, заканчивая уведомлениями в реальном времени о событиях на дороге.

Существующие СОИМ (VEMA (Чехия), Verkehr, aim4it (Германия) и другие) предлагают ограниченный набор ИТС и, как правило, создаются для конкретного региона, что затрудняет их расширение. Некоторые компании (например, Deutsche Bahn), предоставляют ИТС, сохраняя монопольное право на использование информации, что позволяет предоставлять к ней эксклюзивный доступ и возможность монетизации, однако затрудняет использование ИТС в СОИМ. Таким образом, разработка открытой СОИМ, позволяющей повысить мобильность населения за счет использования открытых данных о транспорте, открытых информационных транспортных и геоинформационных сервисов, является актуальной и востребованной задачей.

**Целью диссертационной работы** является разработка методов и моделей для построения сервис-ориентированной СОИМ, позволяющей повысить качество обслуживания пользователей за счет использования открытых данных и сервисов как при планировании поездки, так и во время нее. Для достижения поставленной цели в работе решены следующие основные задачи:

- 1) Анализ имеющихся исследований по обеспечению инфомобильности, определение характеристик СОИМ, формулировка требований к ним и принципов их построения.

- 2) Разработка подхода к построению СОИМ, учитывающего сформулированные требования и включающего в себя сценарную, концептуальную и онтологическую модели системы.
- 3) Обеспечение семантической интероперабельности сервисов, входящих в состав разрабатываемой СОИМ, и модификации их поведения в зависимости от изменений в системе.
- 4) Разработка метода планирования мультимодальных маршрутов, предоставляющего возможность планирования мультимодальных поездок на локальном, региональном, национальном и международном уровнях с использованием общественного транспорта (далее — ОТ), и возможность планирования совместных поездок с использованием личного транспорта.
- 5) Апробации предложенных решений путем создания прототипа СОИМ.

**Методы исследования.** Для решения поставленных задач в работе используются методы теории множеств, управления онтологиями, управления контекстом, системного анализа, защиты конфиденциальных данных, геопоиска и формирования рекомендаций.

**Положения, выносимые на защиту:**

- 1) Контекстно-управляемый подход к построению системы обеспечения инфомобильности с использованием открытых данных о транспортной инфраструктуре, открытых информационных транспортных и геоинформационных сервисов.
- 2) Сценарная, концептуальная и онтологическая модели сервис-ориентированной системы обеспечения инфомобильности.
- 3) Модификация метода автоматического сопоставления онтологий за счет использования технологии краудсорсинга.
- 4) Метод планирования мультимодальных маршрутов с учетом расписания движения общественного транспорта и возможностью планирования совместных поездок с использованием личного автотранспорта.
- 5) Комплекс программных средств, реализующий предложенный подход, модели и методы для обеспечения инфомобильности.

**Научная новизна** работы состоит в следующем:

- 1) Разработан контекстно-управляемый подход к построению сервис-ориентированной СОИМ, отличающийся использованием независимых друг от друга сервисов, взаимодействующих между собой для выработки общего решения

на основе открытых данных о транспортной инфраструктуре, открытых информационных транспортных и геоинформационных сервисов.

- 2) Предложены сценарная, концептуальная и онтологическая модели сервис-ориентированной СОИМ, на основе которых была разработана архитектура СОИМ, отличающиеся использованием модели «классная доска» для обеспечения асинхронного взаимодействия независимых сервисов, представлением знаний с помощью онтологии для организации взаимодействия сервисов и поддержкой сервисами способности к самоконтекстуализации для адаптации к изменениям в системе.
- 3) Предложена модификация метода сопоставления онтологий, отличающаяся использованием технологии краудсорсинга для автоматизированного сопоставления онтологий с целью обеспечения семантической интероперабельности сервисов системы.
- 4) Разработан метод планирования мультимодальных маршрутов, отличающийся использованием мультиграфа с динамически задаваемыми весами для учета расписания движения общественного транспорта и возможностью планирования совместных поездок с использованием личного автотранспорта.
- 5) Создан комплекс программных средств для обеспечения инфомобильности в сфере туризма, отличающийся использованием сервисов, способных к самоконтекстуализации (планирования маршрутов, поиска объектов на карте и информации о них, выработки рекомендаций), а также сервисов для обеспечения конфиденциальности информации пользователей и сопоставления онтологий.

**Обоснованность и достоверность** научных положений, основных выводов и результатов диссертационной работы обеспечивается анализом состояния исследований в проблемной области, согласованностью теоретических выводов с результатами экспериментальной проверки моделей на наборе сценариев, разработанных в ходе выполнения работы, а также апробацией основных теоретических положений диссертации в печатных трудах и докладах на международных научных конференциях.

**Практическая ценность работы.** Предложенные методы и модели построения сервис-ориентированных СОИМ могут быть использованы для разработки систем данного класса, внедрение которых на локальном, региональном и национальном уровнях позволит обеспечить пользователей программным средством, значительно упрощающим планирование поездок и предоставляющим информационную поддержку в ходе поездки.

**Реализация результатов работы.** Исследования, отраженные в диссертации, проведены в рамках НИР: грантов РФФИ № 13-07-12095 «Методы и модели поддержки социально-ориентированных решений участников транспортного процесса» (2013-2015 гг.), 13-07-00336 «Контекстно-ориентированное управление знаниями для поддержки принятия решений участниками интеллектуального пространства» (2013-2015 гг.), 13-01-00286 «Разработка онтологических моделей и механизмов контекстно-зависимого доступа к ресурсам интеллектуального пространства» (2013-2015 гг.); гранта КА322 «Разработка трансграничной туристической информационной инфраструктуры (Smart e-Tourism)» (2012-2014 гг.) программы ENPI в республике Карелия, финансируемого совместно Европейским союзом, Российской Федерацией и республикой Финляндия. Работа также выполнена при государственной финансовой поддержке ведущих университетов Российской Федерации (субсидия 074-U01). Результаты, полученные в ходе исследования, применяются в сервисе рекомендации достопримечательностей в республике Карелия, в системе управления группой роботов ЦНИИ РТК, а также в учебном процессе по курсу «Интеллектуальные системы и технологии» кафедры информационных систем Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (Университет ИТМО) и в учебном процессе по курсу «Сервис-ориентированные системы» Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».

**Апробация результатов работы.** Результаты диссертационного исследования представлялись на международных конференциях ассоциации открытых инноваций FRUCT: FRUCT 11 (Санкт-Петербург, 2012), FRUCT 13 (Петрозаводск, 2013), FRUCT 14 (Хельсинки, 2013), FRUCT 15 (Санкт-Петербург, 2014), FRUCT 16 (Оулу, 2014; лучший демо-стенд), международных конференциях по интеллектуальным пространствам “tuSMART/NEW2AN” (Санкт-Петербург, 2012, 2014), международном семинаре “Information Fusion and Geographic Information Systems (IF&GIS)” (Санкт-Петербург, 2013), международной конференции “Cyber Conflict (CyCon)” (Таллин, 2013), международной конференции “Enterprise Information Systems (ICEIS)” (Анже Луар, 2013), международной конференции “Joint conference on Software Technologies (ICSoft)” (Рейкьявик, 2013), международной конференции “Knowledge Engineering and Semantic Web (KESW)” (Казань, 2014; лучшая статья конференции). По разработанным сервисам также были получены свидетельства о регистрации программ для ЭВМ «Система обеспечения безопасности личных данных в интеллектуальных пространствах» № 2015619362 от 01.09.2015 и «Сервис рекомендации достопримечательностей для системы обеспечения инфомобильности» № 2015619363 от 01.09.2015.

**Публикации.** По материалам диссертации опубликовано 28 печатных работ, включая три работы в журналах из списка ВАК («Труды СПИИРАН» и Journal of Intelligent Systems) и 11

работ в международных изданиях, индексирующихся в реферативных базах Web of Science и Scopus.

**Структура и объем работы.** Диссертация объемом 186 машинописных страниц, содержит введение, четыре главы и заключение, список литературы (180 наименований), 7 таблиц, 50 рисунков, одно приложение с копиями актов внедрения.



# Глава 1. Инфомобильность. Требования к построению системы обеспечения инфомобильности

## 1.1 Интеллектуальная транспортная система. Понятие инфомобильности

В настоящее время в Российской Федерации наблюдается устойчивый рост обеспеченности населения индивидуальными автомобилями. Согласно данным федеральной службы государственной статистики, с 2000 по 2013 годы количество легковых автомобилей в собственности граждан увеличилось на 105 % — с 19 млн. до 39,2 млн. (см. рисунок 1). В период с 2012 по 2013 год, прирост в среднем по стране составил 6 % — с 36,9 млн. до 39,2 автомобилей [23]. Следует, однако, отметить, что в настоящее время по количеству автомобилей на 1000 человек Россия значительно отстает от стран Западной Европы и США, в которых этот показатель колеблется от 500 до 900 автомобилей на 1000 человек [168], против 273,1 автомобиля на 1000 человек для Российской Федерации.

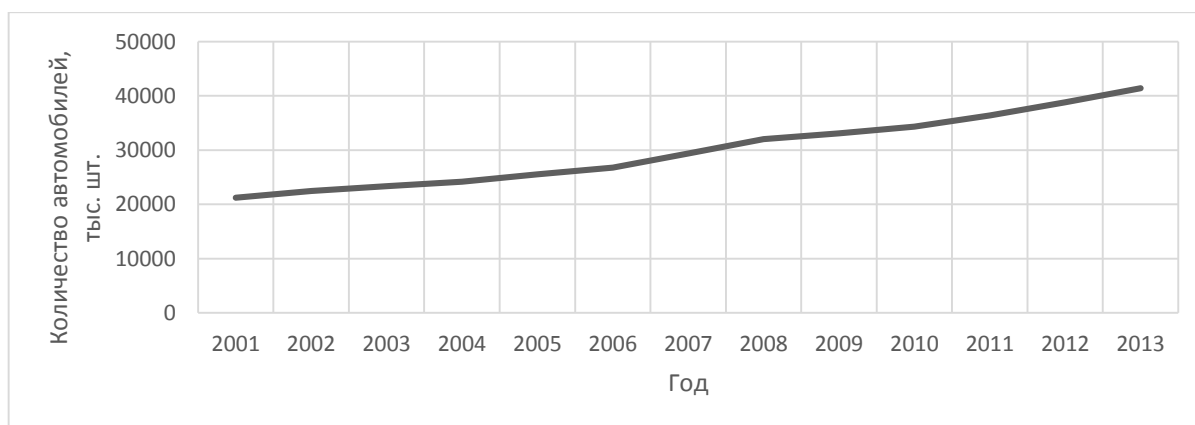


Рисунок 1 — Динамика роста количества личного автотранспорта в Российской Федерации за 2001-2013 гг.

Кроме роста количества личного автотранспорта, наблюдается процесс активной урбанизации. За период с 2002 по 2013 год при незначительно изменившемся соотношении городского и сельского населения в пользу городского - 73 % городского населения к 27 % сельского в 2002 году и 74% городского населения к 26% сельского в 2013 году – значительно возрос процент населения в городах с населением более миллиона человек. В результате миграционного прироста, количество таких городов выросло с 13 в 2002 году до 15 в 2014, и на текущий момент в них проживает 22,5% населения Российской Федерации или 30% всего городского населения [15]. Прирост городского населения, вместе с сохраняющейся средней численностью личных автомобилей на 1000 человек свидетельствует о росте количества транспорта на улицах городов.

Вместе с тем, наблюдается постепенный отказ населения от услуг общественного транспорта. По всем видам наземного городского транспорта падение пассажиропотока с 2000 по 2013 годы составило 57,6%. Вместе с падением пассажиропотока выявлено сокращение количества общественного транспорта на 29 % — со 145,6 тыс. шт. по состоянию на 2000 год, до 102,6 тыс. шт. по состоянию на 2013 год [23].

Приведенная статистика наглядно показывает тенденцию роста количества автотранспорта вместе с падением интереса к услугам общественного транспорта в Российской Федерации. С последствиями данного процесса уже столкнулись многие крупные города: возросло количество заторов, возник дефицит парковочных мест, стало существенно сложнее использовать общественный транспорт. Решение этих проблем должно иметь комплексный характер и проводиться на всех уровнях государственного управления.

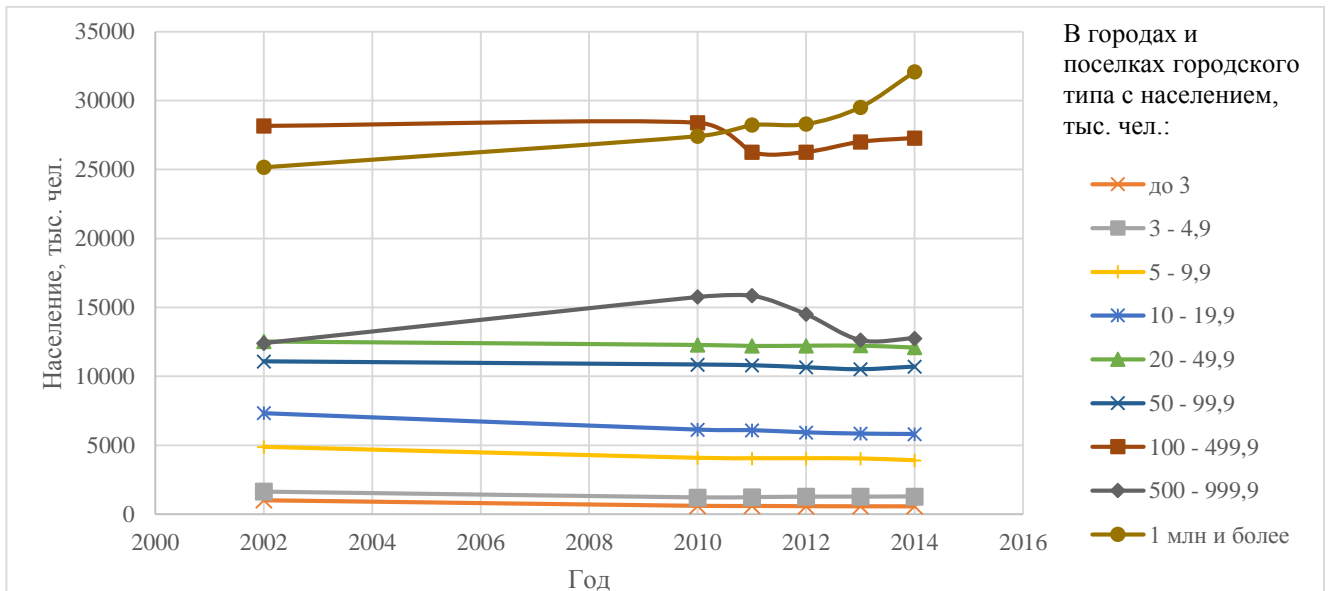


Рисунок 2 — Динамика распределение городского населения в Российской Федерации за 2002-2014 гг.

В ходе анализа проблем развития городской среды, учеными было выведено понятие города, удобного для жизни. Одной из характеристик такого города является «возможность перемещаться с удобствами без необходимости владеть автомобилем или пользоваться им» [1]. Другими словами, данный тезис можно выразить следующим образом: чем удобнее город для автомобилей, тем он менее привлекателен для людей.

Таким образом, можно сделать вывод, что развитая сеть общественного транспорта и наличие удобной пешеходной инфраструктуры являются наиболее важными характеристиками городов, удобных для жизни [1]. Выделяется четыре уровня планирования, организации и функционирования систем городского транспорта [1]:

- 4 уровень. Включает в себя отдельные объекты инфраструктуры, такие как перекрестки, бульвары, пешеходные зоны, отдельные маршруты автобусов и т.д.

- 3 уровень. Включает в себя маршрутную сеть одного вида транспорта: уличная сеть, велосипедные дорожки, сеть пригородных железных дорог.
- 2 уровень. Рассматривается интермодальная координированная система, в состав которой входит сеть улиц и дорог, различные виды общественного транспорта, пешеходные зоны и т. д.
- 1 уровень. Рассматривается город и его взаимосвязь с транспортной системой. На этом уровне осуществляется координация транспортной системы города и его объектов с остальными функциями, такими как экономика, жилищная сфера, социальные условия. Является самым высоким уровнем интеграции процессов планирования и управления.

Обычно, деятельность планирующих и эксплуатационных государственных органов на уровнях «4» и «3» поставлена вполне удовлетворительно [1]. Второй уровень находится за пределами ответственности какой-либо одной транспортной службы, предполагая более высокую степень организации, которая чаще всего возлагается на региональный орган управления. Однако на практике решения многих проблем до сих пор принимаются без рассмотрения взаимодействия между различными видами транспорта. Первый уровень является высшим уровнем координации городского планирования и развития. На нем осуществляется планирование транспорта как единой функциональной системы, связанной с другими аспектами функционирования города, такими как расселение, экономическая и социальная активность, экология и т.д. Вследствие недостаточно детальной проработки второго уровня, деятельность на первом уровне в настоящее время также является неэффективной.

Для повышения эффективности управления городским транспортным планированием на втором уровне необходим всесторонний анализ существующей транспортной сети города, включающей в себя все имеющиеся виды транспорта и существующую улично-дорожную сеть. Комбинация видов транспорта, используемого в конкретном городе во многом зависит от его размера и особенностей. Для малого города выбор видов транспорта чаще всего не является особой проблемой. Большинство потребностей жителей города могут быть удовлетворены за счет индивидуальных перемещений — пешком, на велосипеде или на автомобиле. Для городов с низким уровнем автомобилизации или с наличием большого количества жителей должны развиваться услуги общественного транспорта. Таким образом, малые города с высоким уровнем автомобилизации — это единственное место, где одномодальную систему «автомобиль-дорога», дополненную инфраструктурой для пешеходов, можно считать адекватной и эффективной.

При росте размеров города возрастает потребность в общественном транспорте, обладающем значительными провозными возможностями. Таким транспортом могут быть вместительные автобусы, троллейбусы, сеть городского электрического железнодорожного

транспорта (трамвай, метро). Одновременно возникает ряд проблем, связанных с использованием автомобилей в черте города, чрезмерным потреблением доступных территориальных ресурсов города и негативными внешними эффектами. Поэтому такие города нуждаются во внедрении сбалансированной транспортной системы.

В настоящее время, в целях смягчения существующих транспортных проблем в городе и для решения основных задач второго уровня управления городской транспортной инфраструктурой, а также для повышения мобильности жителей города, многие специалисты в транспортной сфере сосредоточили свое внимание на разработке интеллектуальных транспортных систем (ITS, Intelligent transportation system, далее — ИТС). Согласно директиве Европейского парламента и совета от 7 июня 2010 года, под интеллектуальными транспортными системами понимаются такие системы, в которых информационные и коммуникационные технологии применяются в области дорожного транспорта, включая инфраструктуру, транспортные средства и пользователей, и в управлении транспортом и мобильностью, а также для связи различных видов транспорта [68].

Основной целью данных систем является предоставление конечным потребителям инновационных сервисов, основанных на определении местоположения (LBS, Location-based services), связанных с различными типами транспорта и управлением движением для обеспечения информирования о текущей дорожной ситуации, что позволяет пользователям подобных сервисов совершать более безопасные поездки и эффективнее использовать существующие транспортные сети региона. ИТС интегрируют телекоммуникационные, электронные и информационные технологии с целью планирования, разработки, поддержания и управления транспортными системами города. Применение информационных и коммуникационных технологий в сфере дорожного транспорта и его связь с другими типами транспорта вносит существенный вклад в повышение эффективности использования транспорта и окружающей среды, увеличивает безопасность дорожного движения, а также значительно увеличивает мобильность пассажиров и грузов [68].

Развитие ИТС способствует решению задач в нескольких приоритетных областях, среди которых можно выделить следующие:

- 1) Оптимальное использование дорог, оптимизация транспортных потоков и обмен данными о поездках. Обеспечивая общий доступ к актуальной и точной информации о дорогах, а также информации о распределении трафика и событиях на дорогах, ИТС позволяют произвести перераспределение трафика по менее используемым дорогам, оптимизируя, тем самым нагрузку на существующую дорожную инфраструктуру.

- 2) Непрерывность транспортного потока и управление перевозкой грузов с помощью ИТС. В этой области приоритетной целью является обеспечение интероперабельности различных ИТС, обеспечивающее непрерывность предоставления услуг и мультимодальный способ передвижения, включая единые билеты на несколько видов транспорта и мультимодальные грузоперевозки на национальном, региональном и локальном уровнях.
- 3) Обеспечение сохранности дорожной инфраструктуры и безопасности дорожного движения с помощью ИТС. Данная область включает в себя охраняемые парковки для грузовиков и коммерческого транспорта, контроль за состоянием дорог, обеспечение обмена данными между транспортными средствами и центрами экстренной помощи.
- 4) Связь транспортных средств с дорожной инфраструктурой. Решение задач этой области позволяет интегрировать различные ИТС с различными открытыми программными платформами на транспортных средствах, что в дальнейшем позволит создавать системы кооперации автомобиль-автомобиль, автомобиль-инфраструктура и инфраструктура-инфраструктура, позволяющие решать задачи оптимизации транспортных потоков и обеспечения безопасности поездок.

К ИТС относится широкий спектр систем, начиная от простых систем автомобильной навигации (Навител, СитиГид и т. д), предоставляющих возможность построения маршрута с учетом заданных ограничений, таких как пробки, перекрытия дорог, объезд ДТП и т.д.; регулирования грузоперевозок, позволяющих произвести планирование перевозки грузов между заданными точками с возможностью мультимодальных перевозок, при которых доставка груза до точки назначения производится с помощью нескольких типов транспортных средств; и заканчивая автоматизированными системами управления дорожным движением (АСУДД, например, «Старт» [149], «Спектр» [6], «Магистраль» [166] и т. д.) и системами, интегрирующими информацию из большого количества различных источников с целью моделирования (например, PTV-Optima [138]) и прогнозирования дорожной ситуации (например, Traffic prediction tool, разработанная компанией IBM или SCATS (Sydney Coordinated Adaptive Traffic System) [152], разработанная в агентстве дорог и морских путей Австралии и адаптированная для многих городов мира).

Диссертационная работа нацелена на разработку методов и моделей для построения сервис-ориентированной системы обеспечения инфомобильности, позволяющей повысить качество обслуживания пользователей за счет использования открытых данных и сервисов как при планировании поездки, так и во время нее.

Понятие «инфомобильность» относится к ИТС, предоставляющим пользователям мобильных устройств доступ мультимодальной, динамической, персонализированной информации и транспортным сервисам с учетом их текущего географического положения и контекста [68; 120]. Под контекстом понимается любая информация, которая может быть использована для описания ситуации, в которой находится в данный момент некоторый объект (человек, объект или какое-либо транспортное средство) [66]. Основная цель таких систем заключается в повышении мобильности пользователей за счет информационной поддержки, осуществляемой как при планировании поездки, так и во время нее [38]. В связи с этим, системы, обеспечивающие инфомобильность, предоставляют достаточно широкий набор сервисов для работы с геоинформацией, среди которых можно выделить:

- Отображение карты. Карта может подгружаться из внутренней памяти используемого устройства или из внешних источников и должна быть обновляемой, для предоставления пользователю актуальной информации. Учитывая мобильность пользователя, карта должна охватывать весь мир и иметь высокую детализацию.
- Поиск и отображение точек интереса (POI). Среди точек или мест интереса выделяются достопримечательности (памятники, музеи, мемориальные доски и т.п.), магазины, кафе, заправки и многое другое. Дополнительно, для каждого типа точек интереса может выводиться контекстная информация, например, время работы музея, описание достопримечательности, фотографии пользователей с этого места, оценки пользователей и др.
- Построение и отображение маршрутов. Маршруты могут строиться как для всех видов транспортных средств по отдельности, так и для различной их комбинации (мультимодальный маршрут).
- Отображение событий на карте. Заторы, ДТП, ремонты дороги и прочая актуальная информация, позволяющая скорректировать планируемый маршрут.
- Поддержка пользователя в поездке. Отображение расписания общественного транспорта, навигационная информация, оповещения о событиях.
- Добавление и редактирование информации самими пользователями. Пользователи могут самостоятельно редактировать уже имеющуюся информацию об объектах в системе, либо создавать новые объекты, связанные с определенным географическим положением.
- Дополнительные сервисы, такие как погода, бронирование отелей, прокат машин, наличие свободных мест на парковке, стоимость бензина на заправочных станциях по бренду и др.
- Мобильность. Система должна полностью или частично (основная обработка информации производится в облаке, устройство пользователя используется только для

сбора данных и отображения результата) работать на мобильном устройстве конечного пользователя.

- Учет предпочтений пользователя. Вся предоставляемая информация должна учитывать предпочтения пользователя.
- Рекомендации. Предоставляемая информация может быть дополнительно ранжирована, основываясь на отзывах других пользователей и близости их интересов к интересам пользователя системы.

Существующие СОИМ (VEMA (Чехия), Verkehr, aim4it (Германия) и другие) предлагают ограниченный набор ИТС и, как правило, создаются для конкретного региона, что затрудняет их расширение. Некоторые компании (например, Deutsche Bahn), предоставляют ИТС, сохраняя монопольное право на использование информации, что позволяет предоставлять к ней эксклюзивный доступ и возможность монетизации, однако затрудняет использование ИТС в СОИМ. Таким образом, разработка открытой СОИМ, позволяющей повысить мобильность населения за счет использования открытых данных о транспорте, открытых информационных транспортных и геоинформационных сервисов, является актуальной и востребованной задачей.

## **1.2 Обзор исследований и проектов в области обеспечения инфомобильности**

Несмотря на новизну термина «инфомобильность», первые упоминания которого относятся к началу двухтысячных годов [53; 113; 119], на данный момент существует множество исследований, направленных на разработку систем, обеспечивающих инфомобильность, архитектур для них и требований к подобным системам. Кроме того, некоторые из систем, разработанных в рамках ИТС, также можно отнести к системам, обеспечивающим инфомобильность пользователей.

Как уже упоминалось ранее, задачей обеспечения инфомобильности является предоставление пользователям информации, полезной для совершения передвижения, которая может включать в себя навигацию, мультимодальное планирование маршрута и перепланирование (в том числе услуги по бронированию транспортных средств), предоставление информации на основе определения местоположения, оповещения о событиях, которые могут повлиять на мобильности пользователя. Предоставление данных видов информации может осуществляться как при планировании поездки, так и во время нее. Подводя итог, при обеспечении инфомобильности должны быть решены следующие основные задачи: предоставление навигационной информации, направленной на обеспечение мобильности конкретного пользователя, интеграция информации о мультимодальных перевозках, предоставление контента на основе определения местоположения и поддержка мультимодальных каналов связи.

В обзоре, представленном далее рассмотрены три основные категории по которым разделены существующие исследования:

- 1) Персональные навигационные системы: сервисы для навигации по пешеходным маршрутам, маршрутам общественного транспорта или частных транспортных средств;
- 2) Среда для интеграции инфомобильности: системы разработанные, для интеграции различных информационных ресурсов обеспечения мобильности;
- 3) Контекстно-зависимые платформы на основе определения местоположения связи для сетей следующего поколения.

### **Персональные навигационные системы**

#### *Personal Travel Companion*

В работе [141] представлен прототип, названный *Personal Travel Companion*, направленный на обеспечение помощи пользователям, включающей персонализированное мультимодальное планирование поездки, а также мобильное управление мультимодальной поездкой в ходе поездки. Прототип обеспечивает планирование мультимодального маршрута «от двери до двери» с настольного ПК или мобильного устройства. Маршрут рассчитывается по интегрированным пешеходным, дорожным сетям и сети общественного транспорта. Пользовательские настройки (например, предпочитаемые виды транспорта, требования к мобильности, возможности и ограничения по времени) учитываются в системе для расчета персонализированных маршрутов. Запланированные маршруты могут быть сохранены для ведения личной истории маршрутов.

Оказание поддержки в ходе поездки осуществляется посредством мультимодальных расширений в навигационной системе автомобиля или персонального ассистента на мобильном устройстве пользователя. Непрерывность и цельность поддержки обеспечивается средствами мобильного управления мультимодальной поездкой на различных устройствах и навигации и ориентации к адресу назначения (включая пешеходную навигацию при пересадке). Отклик в реальном времени не гарантирован и зависимость от контекста задумана только с точки зрения адаптации содержания услуг к местоположению пользователя, его предпочтениям и имеющимся транспортным средствам. Доступ к сервисам основан на использовании GPRS, таким образом, бесшовность всегда обеспечена коммуникационной платформой и специально не рассматривается в данной работе. Прототип не включает в себя механизмы и методы, направленные на обеспечение требований достоверности, конфиденциальности и безопасности информации.

#### *Smart Travel Information Service (STIS)*



STIS [50] — это система, предлагающая путешественникам сервис мультимодального планирования путешествия, направленный на преодоление разрыва в координации имеющихся транспортных систем (Бреннан и Мейера, 2007). План путешествия создается на основе предпочтений, высказанных конечными пользователями, и интегрирует статическую и динамическую информацию о пробках и общественном транспорте.

STIS является связующей платформой, использующей данные, предоставленные средой iTransit, и использует интерфейс HTTP, чтобы предоставить пользователям персонализированные планы путешествия. Сервис может быть доступен с персонального компьютера ПК или мобильного телефона, что осуществляется с помощью обмена XML сообщениями по протоколу HTTP. Среда iTransit объединяет информацию, поступающую из различных наследованных систем, основанных на общей пространственной модели данных. В качестве контекстной информации здесь используется информация о трафике и информация об общественном транспорте, в то время как использование контекста пользователя (например, его расположение) не является основным. Кроме того, в основное внимание сосредоточено на планировании поездки, а не на помощи во время нее. Информация в режиме реального времени с учетом условий мобильности может быть получена путем интеграции мониторинга дорожного движения и транспорта через среду iTransit, но система не поддерживает динамическую перепланировку маршрутов пользователя, основанной на информации в реальном времени, такой как задержки общественного транспорта и пробки.

### *Navitime*

Navitime является мобильным навигационным сервисом, предлагаемым в Японии и используемым большим количеством пользователей (1,82 млн пользователей по оценкам на январь 2007) [40]. Navitime предоставляет пользователям расчет пути и услуги ведения вдоль маршрута, объединяя несколько видов транспорта: пешеходные маршруты, личный автотранспорт, поезда, такси и самолеты. Navitime представляет собой распределенную систему: серверы вычисляют маршруты и генерируют карты, мобильные клиенты собирают информацию о местоположении, обрабатывают информацию, вводимую пользователями, загружают данные и визуализируют карты. Маршруты выбираются на основе ряда критериев, среди которых: самый быстрый маршрут, минимизация расходов на поездку, длина пешеходного маршрута. Контекстная информация, которая управляется системой включает в себя местоположение пользователя, полученное через устройства GPS или ручную указанное конечными пользователями и используется для адаптации навигации и помощи в текущем местоположении, транспортная информация и другая информация с географической привязкой, используется для пометки карт и маршрутов (например, дополнительная информация о погоде, количество выделяемого углекислого газа для каждого маршрута).

Navitime объединяет информацию о мобильности предоставленной гетерогенными источниками данных путем преобразования их в четыре распространенных формата: Dformat для данных расписания, Mformat для дорожной сети данных, Vformat для 2D представления картографических данных и V3Dformat для 3D представления картографических данных. Публично доступная документация не имеет прямых ссылок на соблюдение Navitime существующих стандартов хранения и обмена информации о мобильности.

### **Среда для интеграции инфомобильности**

#### *iTransit*

iTransit является основой для многоуровневой архитектуры интеллектуальной транспортной системы, разработанной для интеграции современных и традиционных интеллектуальных транспортных систем [50]. iTransit основан на многоуровневой модели данных объекта. Взаимодействие достигается путем указания этой общей модели данных в качестве основы федеративной архитектуры гетерогенных и устаревших интеллектуальных транспортных систем. Модель содержит пространственные и временные аспекты данных перевозок и движения и представляет собой единый механизм для осуществления запросов и обработки информации, поступающей от гетерогенных транспортных систем. Они также включают в себя глобальные информационные уровни (т.е. данные, представляющие общий интерес), содержащие физическую и политическую географию и транспортную сеть региона. Глобальные уровни могут быть расширены с помощью системы слоев, которые представляют информацию, созданную или используемую в конкретной системе (например, расписание общественного транспорта, погодная информация, и т.д.).

Взаимодействие между различными уровнями данных обеспечивается набором понятий высокого уровня (названными «Контекстной абстракцией»), представляющими основные абстракции для глобальных и системных уровней: реального мира, системы, данных, расположения и идентификации объектов.

Для этой модели данных был разработан программный интерфейс. Обмен данными между системами, составляющими архитектуру iTransit основан на использовании технологии CORBA и веб-сервисов. Модель данных соответствует спецификации OpenGIS. Построенная на этой модели архитектура iTransit состоит из трех уровней, каждый из которых реализует разный уровень взаимодействия: уровень наследования, включающий унаследованные системы, реализующие свою собственную модель данных; уровень iTransit интегрирующий транспортные системы, изначально реализующие модель данных iTransit, и уровень приложений, включая сервисы с добавленной стоимостью, построенные на системе iTransit.

#### *Highway Traveler Information System in the Jiangsu Province of China*

Xiang др. (2007) [178] описывают принципы и основные моменты проектирования, которые были приняты для реализации информационной системы для поездок по трассам Highway Traveler Information System (HTIS) в провинции Цзянсу Китая. В Китае опубликованы спецификации для определения архитектуры национальной интеллектуальной транспортной системы. Тем не менее, провинция Цзянсу является одной из немногих китайских провинций, которые запланировали развитие HTIS в рамках научно-исследовательских проектов, продвигаемых Транспортным департаментом провинции Цзянсу.

Архитектура HTIS направлена на обеспечение среды, реализующей взаимодействие между распределенными гетерогенными системами. Она организована на семи логических уровнях, представляющих разные политические географические организации, ответственные за обмен данными, сетевую маршрутизацию и наследование систем. Эти уровни включают в себя: провинциальный узловой уровень, размещающий провинциальные узлы, которые представляют централизованный сбор и распределение точек данных и подключенные к различным региональным центрам; региональный узловой уровень, включающий 14 региональных центров взаимодействия с источниками данных и предоставляющий пользователям информацию в каждом регионе; уровень интерфейса концентратора, который поддерживает интеграцию с унаследованными интеллектуальными транспортными системами путем преобразования унаследованных протоколов и форматов данных в спецификации HTIS; уровень подсистемы HTIS, в том числе всех подсистем, которые получают данные из полевых устройств и подключаются к своим региональным центрам (непосредственно или через хаб интерфейсов); уровень полевых устройств, включая устройства (например датчики, детекторы и т.д.), получающие исходные данные для мониторинга дорожного движения; уровень поставщиков информационных сервисов включающий основанных на информации агентства, которые подключаются к провинциальным узлам и используют данные HTIS для распространения конечным пользователям (через киоски, персональные устройства пользователей, и т.д.) и анализа; уровень сети Интернет, включающий в себя сервисы, обеспечивающие конечных пользователей данными HTIS через Интернет.

Основное внимание при сборе контекстной информации уделяется информации, касающейся мониторинга и контроля за шоссе, в то время как контекстные элементы, сосредоточенные на конечных пользователях (например, их расположение, активность и т.д.), не принимаются во внимание. Акцент делается на модели интеграции разнородных и распределенных информационных систем, основанных на многоуровневой архитектуре, общих протоколах и форматах данных.

*Arktrans*

Arktrans является средой для мультимодальных интеллектуальных транспортных систем в Норвегии [124]. Его целью является создание общей точки зрения на проблемную область транспорта для всех видов транспорта (автомобильный, морской, железнодорожный и воздушный) в терминах стандартных функций и интерфейсов для взаимодействия между разнородными транспортными системами.

В настоящее время, технические характеристики Arktrans определяют информацию, необходимую пользователям, путешествующим с использованием общественного транспорта, такую как расписания, информация о задержках и услуги в режиме реального времени, предоставляемые на борту транспортного средства и на остановках. Эта информация определяется концептуальной моделью, представляющей понятия и логические отношения, определенные посредством UML-диаграмм и синтаксиса XML. Существующие европейские стандарты, такие как TRANSMODEL, были проанализированы, но не реализованы, так как они не полностью решают некоторые основные требования Arktrans (например, мультимодальность с особым акцентом на использовании водного транспорта). В настоящее время разработано несколько пилотных проектов, связанных с созданием планировщика мультимодальных поездок «от двери до двери», сочетая планирование по расписанию (с использованием общественного транспорта, автомобильным, морским и железнодорожным транспортом) и нерегулярных (автомобиль, велосипед, пешком) транспортных средств. Расчет персонализированного маршрута производится с учетом предпочтений пользователя. Работа системы сосредоточена на поддержке пользователей при планировании поездки, в то время как интеграция навигационного сервиса, не осуществлена.

**Контекстно-зависимые платформы на основе определения местоположения***Lol@*

Lol@, помощник в локальном расположении, является прототипом туристического гида, работающего через UMTS, который был разработан в рамках совместного проекта в исследовательском центре телекоммуникаций Вены [172]. Lol@ является сервисом для экскурсий разработанным на архитектуре для сервисов на основе определения местоположения, реализующих стандарты SIP (Session Initiation Protocol) и OSA/Parlay. Он обеспечивает туристов услугами навигации по предопределенным маршрутам, включая достопримечательности и услугами доставки мультимедийного контента, связанного с этими достопримечательностями.

Приложение основано на клиент-серверной архитектуре и реализовано с использованием Java-апплетов и Java-приложений вместе с XML-технологией для представления контента. Связь с сервером осуществляется по протоколу HTTP через GPRS или UMTS. Lol@ обеспечивает функции навигации и маршрутизации в центре города Вена. Точки интереса, и маршруты

показываются на карте, и пользователи могут взаимодействовать с приложением, путем нажатия на иконки и гипертекстовые ссылки или через голосовые команды. Пользователи могут выбрать между текстовой, графической и голосовой навигацией.

Контекст моделируется с точки зрения местоположения пользователя. Lol@ использует различные методы позиционирования: GPS, позиционирование на основе мобильной сети и ручной ввод местоположения пользователями. Сервер позиционирования повышает точность информации о местоположении, получая информацию о местоположении от GPS или инфраструктуры сотовой сети оператора и рассчитывая точное местоположение. Это используется для расширения контекстной модели ссылкой на дату и время (например, часы работы для музеев).

### *MapWeb*

MapWeb является платформой, основанной на определении местоположения и построенной на архитектуре IP Multimedia Subsystem (IMS) [96]. Архитектура MapWeb включает в себя три уровня: уровень приложений, уровень управления и уровень медиа. Уровень приложений содержит размещение сервера MapWeb, который предоставляет информацию на основе местоположения и коммуникационные услуги по интеграции различных серверов приложений, например, серверов расположений, сервера присутствия, и других внешних серверов (например, серверы карт).

Уровень управления обеспечивает сессии и управление маршрутизацией. Уровень медиа включает терминалы, которые иницируют и завершают сеанс связи. Например, приложение MapWeb может показывать интерфейс на основе карты на пользовательском устройстве. Через взаимодействие на основе карты, пользователь может просматривать собственную позицию, найти приятелей и связаться с ними с помощью телефонного звонка или мгновенного обмена сообщениями и найти точки интереса. Система может адаптировать услуги и используемые каналы связи в соответствии с предпочтениями пользователя и его профилем. Кроме того, пользователь может определить правила для адаптации услуг в зависимости от контекста. В MapWeb, контекстом является расположение, текущее время, наличие пользователя и его деятельность (полученная из календаря пользователя). MapWeb конкретно не нацелены на обеспечение инфомобильности; а обеспечивает платформу, позволяющую производить разработку таких сервисов, основанных на интегрированных сервисах коммуникации, персонализации и управления контекстом.

### *IMS-based location-aware services*

IP-Multimedia Subsystem (IMS) является международным стандартом для расширенного предоставления мультимедийных услуг в конвергентных сетях следующего поколения. Несколько научных исследований направлено на изучение возможностей расширения IMS

поддержкой услуг на основе определения местоположения. Преимущество такого подхода заключается в расширении услуг на основе определения местоположения дополнительными функциями, предусмотренными архитектурой IMS, такими как поддержка качества обслуживания (QoS), интеграции услуг на базе стандарта над IP инфраструктурой.

В работе [122] предлагается расширение архитектуры IMS путем разработки и реализации сервера определения местоположения IMS, что обеспечивает серверы IMS и IMS приложения информацией о месте, полученной из различных систем позиционирования. Сервер местоположения IMS является общим сервером приложения SIP, который может получить информацию о местоположении от системы позиционирования на основе сети с помощью передвижного пункта протокола (MLP) [128]. В дополнение к этой сетевцентричной точке зрения, Fabini др. была предложена система определения местоположения, ориентированная на обеспечение ядра IMS и IMS-сервисов информацией о местоположении, полученной с клиентских устройств, оборудованных системами позиционирования (GPS или, например, Galileo приемниками) [74]. Предложенная методика также взаимодействует с механизмами сетевцентричного местоположения и направлена на повышение точности определения местоположения. Конфиденциальность информации пользователей обеспечена механизмом приватности, основанным на SIP [99].

#### *OnRoute*

В работе [78] описывается платформа OnRoute для разработки мобильных приложений, способных предоставить информационные сервисы, связанные с общественным транспортом. Эти сервисы могут использоваться широким кругом пользователей: путешественники, персонал транспортных компаний и компаний, связанных с управлением дорожным движением. Сервисы, предлагаемые авторами и разработанные с использованием платформы, могут быть доступны даже с маломощных мобильных устройств, что требует, однако, наличия хорошего беспроводного интернета.

Для разработки платформы, авторы определяют следующие требования, которым должна удовлетворять его архитектура:

- Поддержка гетерогенной среды мобильных устройств. Должны поддерживаться мобильные устройства разных типов (телефоны, смартфоны, планшеты) на разных операционных системах.
- Масштабируемость. Система должна позволять добавление новых элементов, предоставляющих новые функции, в инфраструктуру и делать их доступными для широкого круга пользователей.
- Произвольное взаимодействие. Система должна позволять произвольное взаимодействие любых пользователей с любыми сервисами системы.

С точки зрения распределенных систем, авторы характеризуют приложения, разработанные с использованием платформы OnRoute, способностью интегрировать окружающие физическую и технологическую среды. Следовательно, они могут оперировать автономно и произвольно в различном окружении. Для достижения этой функциональности, OnRoute реализует принципы, характерные для распределенных систем. Принцип границы гласит, что разделение между средой и распределенной системой обязано быть сделано по границам, отмечающим разницу в контенте и эти границы не должны ограничивать интероперабельность системы. Принцип волатильности гласит, что распределенная система должна учитывать тот факт, что количество пользователей, устройств и приложений в распределенной системе является непредсказуемым параметром. По этой причине должны быть специфицированы инвариантные принципы работы системы.

Из-за этих характеристик, разработка архитектуры системы OnRoute ведется в двух областях. Первой является инфраструктура, которая обеспечивается системой общественного транспорта. Она включает в себя базовый набор компонентов, содержащий элементы, с помощью которых пользовательские приложения могут получить доступ к информации, связанной с транспортом. Вторая область — пользовательские устройства, включающая в себя все компоненты, которые могут быть интегрированы в различные среды и облегчить доступ к информации, получаемой с помощью информационных сервисов системы OnRoute.

При разработке системы используется онтология, которая предоставляет следующие преимущества: делает систему интероперабельной, снижает сложности разработки, связанные с разнообразием используемых технологий и облегчает взаимодействие между людьми и компьютерными системами.

#### *Im@gine IT*

В работе [119] предлагается система, использующая агентно-ориентированный подход к построению системы обеспечения инфомобильности. Авторы формулируют требования, доказывающие применимость и необходимость использования агентного подхода для организации взаимодействия между сервисами системы.

Среда, в которой работает система, описывается как обслуживание мобильных пользователей (как правило, с помощью мобильного телефона (смартфона), системы его автомобиля, и т.д.) через беспроводные сети на основе различных технологий, таких как GPS и ГИС, предоставляющих маршрутизацию и картографические системы, и различными видами распределенного контента, такими как расписание общественного транспорта, сведения о трафике, и т.д. Это сложная среда мотивирует на создание сочетания простых услуг, с целью предоставления более сложных. В этом контексте, пользователь может, например, найти

маршрут из одного места в другое, увидеть его на карте и иметь возможность добавить его в закладки.

Авторы уточняют, что сервис должен быть мультимодален в том смысле, что пользователь постоянно меняет транспортные средства и поэтому сервис должен поддерживать пользователя по-разному в разное время. Например, когда он находится в машине, оповещения должны быть в аудио-формате, а не иконками на экране.

Кроме того, сервисы могут быть доступны с использованием различных беспроводных технологий, таких как беспроводная локальная сеть или Bluetooth. Подобные сервисы и их применение для помощи пользователю составляют окружающий интеллект.

Пользователи желают иметь доступ к точной, актуальной информации и возможности передвигаться через другие страны. Поэтому, они нуждаются в информации, генерируемой на локальном уровне и распространяемой глобально через сеть сервисов. Эта сеть должна обеспечивать простую интеграцию распределенных провайдеров сервисов, предоставляющих сервисы, которые не обязательно должны быть известны до начала разработки системы. Все эти требования пользователей дают начало совершенно новой области услуг, а именно мобильным, глобальным, персонализированным мультимодальным сервисам обеспечения инфомобильности.

Опираясь на описанные выше идеи, авторы определяют сервисы, требуемые для предлагаемой системы, под названием Im@gine IT. Планируется, что эта система должна быть в состоянии предоставить транспортные решения и туристические услуги от различного набора поставщиков, в соответствии с потребностями пользователя, в сочетании с его профилем и привычками. Если рассматривать требования более детально, то, по мнению авторов, система должна обеспечивать:

- Поддержку различных типов пользователей. Пользователь сам выбирает свой тип и предпочтения (например, турист, пользователь общественного транспорта, водитель и т.д.) и будет получать услуги, учитывающие эти параметры. Будут приняты меры для обеспечения безопасности личной информации пользователя. Кроме того, пользователь может создать несколько профилей (например, житель пригорода (постоянные поездки в город и обратно), турист, бизнес-путешественник, и т.д.) для того, чтобы иметь различные режимы обслуживания.
- Адаптацию услуг согласно шаблонам поведения пользователя. Любая недостающая информация предоставляется на основе профилирования пользователя. Тем не менее, система должна инициировать свои действия только с разрешения пользователя, чтобы избежать его раздражения или удивления.



- Получение запросов от пользователя, его положения (GPS-координаты, если доступны) и предлагать оптимальные транспортные решения, туристические события (выставки, экскурсии) и близлежащие достопримечательности.
- Отслеживание маршрут пользователя и автоматически предоставлять соответствующие события во время путешествия (например, информацию о пробках или чрезвычайных ситуациях на маршруте).
- Поддержку пользователя, при передвижении между странами. Система должна всегда использовать местные услуги в реальном времени, позволяющие достичь большей точности и скорости загрузки.
- Поддержку пользователя во время смены модальности поездки (смены типа транспорта). Во время поездки в машине должны быть доступны аудио сообщения, во время поездки в общественном транспорте выводить в режиме реального времени информацию о следующей остановке, и сообщать название улицы, на которую следует повернуть во время прогулки.

Авторы выбрали агентно-ориентированный подход для системы исходя из следующих дополнительных требований:

- Система должна учитывать, что пользователи и сервисы распределены как географически, так и по времени.
- Должна учитываться неоднородность услуг, устройств и сетей, которые обеспечиваются различными источниками; услуги должны быть адаптированы к профилям пользователей.
- Элементарные услуги и сервисы должны быть согласованы для того, чтобы предоставить пользователю комплексное индивидуальное обслуживание.

Указанные выше требования требуют наличие таких характеристик, как автономия, проактивность, интеллект и взаимодействие, которые являются основной мотивацией выбора технологии многоагентных систем. Агенты могут обладать достаточным интеллектом, чтобы решать различные задачи автономно (например, предоставлять персонализированную поддержку, осуществлять поиск информации о путешествиях, определять местоположение пользователя и т.д.).

Для разработки агентов как для пользовательских устройств (агент персональной поддержки), так и для интеграторов сервисов и платформ провайдеров (агент-брокер, и простые/составные агенты провайдеров соответственно) используется платформа JADE-LEAP. Оповещение об услугах, предоставляемых сервисами системы Im@ge IT осуществляется с использованием RDF, встроенного в платформу JADE, и онтологии, необходимой для

интерпретации содержания сообщений, разработанной с помощью инструмента построения онтологий Protégé.

В работе [53] авторы также отмечают гетерогенность информации, представляемой инфомобильными сервисами. В качестве типичных примеров инфомобильных сервисов приводятся сервисы, связанные с обработкой геоинформации, такие как: предоставление карт местности, прокладка маршрутов для различных типов транспорта, точки интереса. Данные сервисы являются простыми и статичными и не требуют частого обновления. Однако, для продвинутых сервисов крайне важна поддержка динамической географической информации, такой как пробки, ДТП, погодные условия. В подобных сервисах обмен информацией должен происходить как от сервиса к клиенту, так и обратно, что требует повышенного внимания к разработке архитектуры и выбору используемых технологий. Авторы выделяют три основные проблемы, которые должны быть приняты во внимание при разработке архитектуры:

- Необходимо обеспечить адекватную производительность и масштабируемость. Нагрузка на систему, объединяющую несколько сервисов значительно выше, в сравнении с традиционными сервисами обеспечения инфомобильности, за счет взаимодействия пользователей и совместной работы сервисов.
- Архитектура должна гарантировать высокий уровень доступности. Пользователь полагается на инфраструктуру, получая обновленную информацию в режиме реального времени. Как следствие, низкая доступность архитектуры имеет непосредственное влияние на доступность предлагаемой услуги, с потенциальными проблемами для пользователей.
- Наличие персональных услуг, в которых пользователь взаимодействует с данными, имеющими географическую привязку понимают вопросы, связанные с частной жизнью пользователя. Положение пользователя и его идентичность являются конфиденциальными данными, которые не должны быть раскрыты даже в случае возникновения проблемы безопасности в системе, таких, как нападение.

Авторы обращают внимание на то, что при построении системы обеспечения инфомобильности централизованная архитектура, в которой все функции выполняются на одном центральном вычислительном кластере, является неприемлемым решением для представленных задач в силу того, что каждая из функций требует обработки достаточно большого объема данных и существенных затрат вычислительных ресурсов. Кроме того, централизованная архитектура существенно затрудняет масштабируемость и обеспечение поддержки большого количества пользователей. Преимуществом централизованной инфраструктуры является то, что обработка всей информации происходит в единой, замкнутой в себе среде. Это позволяет создать единый защищенный канал, через который будет передаваться вся обрабатываемая информация,

без выхода за пределы среды. Все пользователи изолированы друг от друга, что препятствует получению неавторизованного доступа к их личным данным. В случае распределенной архитектуры необходима отправка и сбор данных сразу для нескольких сервисов, причем часть из них может быть представлена третьей стороной (например, сбор фотографий по заданным координатам из Flickr). В связи с этим, необходима дополнительная проработка механизмов, обеспечивающих сохранность личной информации пользователей системы.

### **1.3 Требования к построению системы обеспечения инфомобильности**

Разработка системы для обеспечения инфомобильности, реализующей все необходимые пользователю функции, требует особого внимания как к реализации ее отдельных функций, так и к их взаимодействию. Для реализации отдельных функций необязательно разрабатывать собственный сервис. На сегодняшний день существует достаточное количество открытых сервисов, предоставляющих информацию, связанную с определенным географическим местоположением, например, погодные сервисы, выдающие текущую погоду по координатам, геолокационные сервисы, позволяющие определить объекты, расположенные вокруг указанных координат и т.д. В связи с этим, основной задачей разработчика системы для обеспечения инфомобильности является объединение различных сервисов в одной системе и удобное предоставление информации пользователю.

В работе [132] выделяются семь требований к системам обеспечения инфомобильности. Эти требования следующие:

1. Непрерывность обслуживания (бесшовная мобильность). Так как пользователь находится в постоянном движении, доступ к сервисам может быть ограничен по ряду причин, основная из которых — изменения в коммуникационной инфраструктуре. Устройство пользователя может переключаться между доступными видами коммуникации (мобильная сеть, Wi-Fi, WiMax и т.д.), что вызывает проблемы с доступом к сервисам. Сервис должен корректно и незаметно для пользователя обрабатывать смену типа подключения и предоставлять тем самым непрерывное обслуживание.
2. Осведомленность в реальном времени о ситуациях в области мобильности. Выполнение данного требования необходимо для предоставления пользователю полной и актуальной информации о передвижении. Такая осведомленность может быть статичной (картографические данные, расписание движения общественного транспорта) и динамической (например, текущие заторы, ДТП, прогноз погоды). Эта информация может применяться для адаптации контента инфомобильного сервиса, например, для динамического изменения маршрута.

3. Поддержка транспортной мультимодальности. Использование различных типов транспорта является ключевым подходом для обеспечения постоянной и безопасной мобильности. Целью достижения оптимального использования существующего транспорта и дорожной инфраструктуры является снижение нынешних темпов перегрузки инфраструктуры, что может быть облегчено путем предоставления точной и актуальной информации о мультимодальных маршрутах. В настоящее время, информация о маршрутах общественного транспорта может быть получена только с при планировании поездки заранее, в то время как большинство мобильных навигационных систем разработаны для использования автомобиля. В настоящее время автомобильные навигационные системы и мультимодальные системы планирования поездки не являются взаимодействующими доменами.
4. Соблюдение стандартов для совместимости и динамичной интеграции информационных источников. Обеспечение работы в реальном времени и надежности мобильных услуг требует интеграции гетерогенной, статической и/или динамической информации, поступающей из различных автономных источников, таких как транспортные операторы, государственные учреждения и центры управления движением. Взаимодействие сервисов и интеграция разнородной информации требуют соглашения об общих правилах и моделях для обмена информацией.
5. Персонализация и адаптация к поведению пользователя. Персонализация определяет то, каким образом информационные службы должны адаптироваться к профилю пользователя и его предпочтениям. В системах обеспечения инфомобильности такой подход должен быть ориентирован на предпочтения пользователей и профили, явно связанные с мобильностью (например, ограниченные физические возможности пользователя или предпочтение частного транспорта). Взаимодействие с пользователем должно быть адаптировано в соответствии с различными параметрами, такими как различные пользовательские устройства, предпочтения, профили и текущие условия.
6. Достоверность информации. Информация, предоставляемая конечным пользователям, должна быть достоверной, или, по крайней мере, должна соответствовать конкретному уровню обслуживания, согласованному с конечными пользователями. Неточная, неполная или неверная информация может поставить под угрозу достижение конечной цели (поддержка пользователей в их мобильности) и, таким образом, вызвать недовольство пользователей системы обеспечения инфомобильности.
7. Конфиденциальность и безопасность. Эффективность контекстно-зависимой системы обеспечения инфомобильности сильно зависит от непрерывного и распределенного сбора и обработки пользовательских конфиденциальных данных, таких как текущая позиция,

активность, ограниченные физические возможности и состояние здоровья. Таким образом, для достижения признания пользователей системы обеспечения инфомобильности требуется обеспечение методов защиты конфиденциальности и личного контроля за доступом к данным. Методы защиты конфиденциальности пользователя могут включать анонимность, криптографию, системы приватности на основе политик и т.д.

Представленные требования были дополнены и расширены после дополнительного анализа работ, связанных с обеспечением инфомобильности [49 – 53; 78; 113; 119; 157] и построением сложных многоагентных систем [4], с целью повышения качества предоставляемых услуг по обеспечению инфомобильности. Список требований, которым должна удовлетворять система обеспечения инфомобильности представлен далее.

*1. Разрабатываемая система должна быть ориентирована на использование открытых данных.*

В настоящее время все больше информации в сети Интернет распространяются в рамках концепции открытых данных, подразумевающей предоставление доступа к информации без ограничений авторского права [18] в структурированном виде, пригодном для машинной обработки. В число таких данных также входят многочисленные виды географической информации, такой как карты, погода, достопримечательности, организации и т. п., и информация о транспортной системе регионов, включая маршруты общественного транспорта, расписания движения транспортных средств на маршрутах, информация о заторах и т.д. Использование открытых данных позволит предоставить пользователю информацию, наиболее актуальную с учетом сложившихся обстоятельств.

*2. Разрабатываемая система должна быть ориентирована на использование сервисов.*

Требование означает, что система должна быть ориентирована на использование открытых информационных сервисов, включающих в себя транспортные сервисы и сервисы, предоставляющие информацию в зависимости от местоположения пользователя (погода, достопримечательности), что требует разработки механизмов для обеспечения их совместной и бесперебойной работы в единой системе, включая механизмы совместного доступа к информации и механизмы обеспечения семантической интероперабельности сервисов.

Данное требование также определяет использование распределенной архитектуры для организации совместной работы сервисов, на что указывают авторы всех работ, представленных в обзоре. Подобный подход делает возможным поддержку большого количества разнообразных устройств и информационных сервисов. Добавление новой функциональности осуществляется запуском нового сервиса, без необходимости существенной переработки системы.

*3. Система должна поддерживать планирование мультимодальных маршрутов и совместных поездок с использованием личного автотранспорта.*

Существующие информационные транспортные сервисы должны использоваться для построения маршрутов по отдельным видам транспорта и обеспечить, при необходимости, смену видов транспорта в ходе следования по маршруту. Выполнение этого требования обеспечит пользователю возможность планировать свои поездки используя несколько видов транспорта. Например, добраться из Санкт-Петербурга в Хельсинки можно несколькими способами: 1) воспользоваться автобусом или метрополитеном, чтобы доехать до вокзала, там пересесть на поезд в Хельсинки; 2) доехать на метро до остановки автобуса в аэропорт, пересесть на автобус, доехать до аэропорта и сесть в самолет до Хельсинки. Поддержка построения мультимодальных маршрутов позволит планировать подобные действия при помощи системы с учетом расписания движения всех транспортных средств.

Планирование совместной поездки с использованием личного транспорта позволит значительно разгрузить существующую дорожную инфраструктуру, снизить затраты на содержание автомобиля и объем топливных ресурсов, необходимых для перевозки определенного количества людей. Все это способствует как улучшению экологической обстановки в городе, так и повышению безопасности и удобства передвижения по региону.

*4. Система обеспечения инфомобильности должна предоставлять информацию пользователям с учетом их предпочтений.*

Основная цель обеспечения инфомобильности заключается в предоставлении конечному пользователю персонализированной информации о способах перемещения и поддержке в процессе перемещения. Должны учитываться особенности пользователя (например, инвалидность), его предпочтения (например, искать маршрут с использованием только электрического транспорта) и т. д.

Предпочтения пользователя могут быть заданы путем создания его профиля, отражающего основные характеристики, присущие пользователю. Благодаря использованию технологии интеллектуальных пространств, предложенных в описании к первому требованию, возможно определение профиля пользователя в виде онтологии. Данный подход в сочетании с сервис-ориентированной архитектурой дает неоспоримое преимущество, связанное с возможностью использования профиля пользователя во всех сервисах. Однако, в подобном подходе есть и отрицательная сторона, связанная с необходимостью обеспечения безопасности конфиденциальных данных пользователя, которые также могут содержаться в профиле для предоставления персонализированной информации. Как было упомянуто в работе [132], методы обеспечения конфиденциальности информации пользователя могут включать анонимность, криптографию, системы приватности на основе политик и т. д.

5. Система должна предоставлять информацию и оказывать поддержку в режиме реального времени.

Поддержание актуальной информации о ситуации на дорогах является важным атрибутом. Своевременное обнаружение затора позволит быстро перестроить маршрут в объезд проблемного участка. Поиск объектов на карте должен предоставлять актуальную информацию о режиме работы объекта. Для этого должны использоваться местные провайдеры услуг, как правило всегда имеющие более высокую точность предоставляемой информации и высокую скорость работы.

6. Система должна учитывать текущую ситуацию при выработке вариантов поддержки пользователя.

В процессе работы система обеспечения инфомобильности должна учитывать текущую ситуацию, для предоставления вариантов поддержки, наиболее подходящих пользователю в определенный момент времени, с учетом текущей ситуации. Например, при обнаружении затора на пути пользователя, система должна найти путь в обход. Другой пример учета текущей ситуации – рекомендация объектов на карте в зависимости от погоды. Например, парк лучше посещать в солнечную погоду, а дождь можно переждать в ближайшем кафе, имеющем много положительных отзывов.

Учет текущей ситуации должен обеспечиваться каждым сервисом в отдельности на основе информации, предоставленной сервисами, описанными в требованиях к четвертому пункту.

7. В системе должна быть обеспечена конфиденциальность информации о пользователе.

В отличие от централизованной архитектуры, распределенная вносит дополнительные задачи, связанные с обеспечением безопасности личной информации пользователей. Преимуществом централизованной инфраструктуры является то, что обработка всей информации происходит в единой, замкнутой в себе среде. Это позволяет создать единый защищенный канал, через который будет передаваться вся обрабатываемая информация, без выхода за пределы среды. Все пользователи изолированы друг от друга, что препятствует получению неавторизованного доступа к их личным данным. В случае распределенной архитектуры необходима отправка и сбор данных сразу для нескольких сервисов, причем часть из них может быть представлена третьей стороной (например, сбор фотографий по заданным координатам из Flickr). В связи с этим, необходима дополнительная проработка механизмов, обеспечивающих сохранность личной информации пользователей системы.

В таблице 1 представлено сравнение систем, упомянутых в обзоре, выполненное на основе выведенных требований к системе обеспечения инфомобильности.

Таблица 1 — Сравнение существующих подходов к обеспечению инфомобильности

Требование	Personal Travel Companion	STIS	Navitime	iTransit	HTIS	Arktrans	Lol@	MapWeb	IMS-based location-aware services	OnRoute	Im@gine IT
Использование открытых данных	Нет	Нет	Нет	Нет	Нет	Нет	Нет	н/д	Нет	н/д	Нет
Тип архитектуры	н/д	Распределенная	Распределенная	Распределенная, многоуровневая	Распределенная	Распределенная	Клиент/сервер	Распределенная	н/д	Распределенная	Распределенная, Агентно-ориентированная
Мультимодальность маршрутов	Да	Да	Да	Да	Да	Да	Нет	Нет	Нет	Да	Да
Учет предпочтений пользователя	Да	Нет	Да	Нет	Нет	Да	Да	Да	Да	Да	Да
Работа в режиме реального времени	Нет	Да	Да	Да	Да	Да	Да	Да	Нет	Да	Да
Учет текущей ситуации	Да	Нет	Да	Да	Да	Да	Нет	Да	н/д	Нет	Да
Обеспечение конфиденциальности пользователя	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Да	Нет	Нет

Для поддержки пользователя во время поездки система должна постоянно контролировать среду, в которой находится пользователь, и адаптироваться к ее изменениям. Среда описывается контекстом, который формируется как самим пользователем, так и различными объектами среды независимыми от него. Контекстной информацией, необходимой для работы инфомобильного сервиса, чаще всего является:



- Тип пользователя в системе (турист, пассажир общественного транспорта, водитель, попутчик и др.). Так как одним из свойств инфомобильных систем является интермодальность маршрута, тип пользователя может динамически изменяться и, следовательно, система должна подстраиваться под каждый из типов (использовать различные виды уведомлений, выводить или не выводить точки интереса и др.);
- Профиль пользователя, который включает в себя личные предпочтения пользователя. Любая дополнительная информация должна выводиться с учетом пожеланий пользователя: музеи определенной направленности, магазины конкретных брендов и т.д.
- Текущие координаты пользователя. Вся информация подбирается с учетом текущего положения пользователя и направлена на его поддержку в текущей точке.
- Текущее время. От времени зависит доступность конкретного транспорта, доступность для посещения музеев, магазинов, и т.д.

Для оказания своевременной поддержки пользователю сбор контекстной информации должен вестись в режиме реального времени, без необходимости дополнительных действий со стороны пользователя. Это позволяет сократить время ожидания результатов и повысить отзывчивость сервиса для пользователя. Кроме реакции на изменение контекста, сервисы могут отслеживать динамику изменения ситуации и, собрав дополнительные данные, предложить пользователю тот сервис, который с наибольшей вероятностью будет полезен в складывающейся ситуации. Такое поведение называется проактивным и в настоящее время разработка многих информационных систем ведется с поддержкой проактивного режима действий [9; 104].

#### **1.4 Технологии, применяемые в проектировании и разработке систем обеспечения инфомобильности**

Анализ существующих исследований позволил выделить три категории информационных ресурсов [27] в системах обеспечения инфомобильности (рисунок 3).

Первой категорией являются ресурсы физической среды. К физической среде относятся параметры, описывающие материальные характеристики объектов, такие как конкретные координаты в зависимости от системы отсчета (обычно широта и долгота), температура воздуха, температура поверхности земли, текущее время и т.п. Отличительной чертой ресурсов физической среды является то, что они могут быть получены непосредственным измерением с помощью физических приборов, таких как термометр, часы, компас и т.п., реализованных в виде физических сенсоров, выступающих как в роли отдельных приборов, так и частью более сложных устройств, например, мобильного телефона. Сбор информации о физических

характеристиках окружающей среды позволяет получить базовый контекст для системы обеспечения инфомобильности.

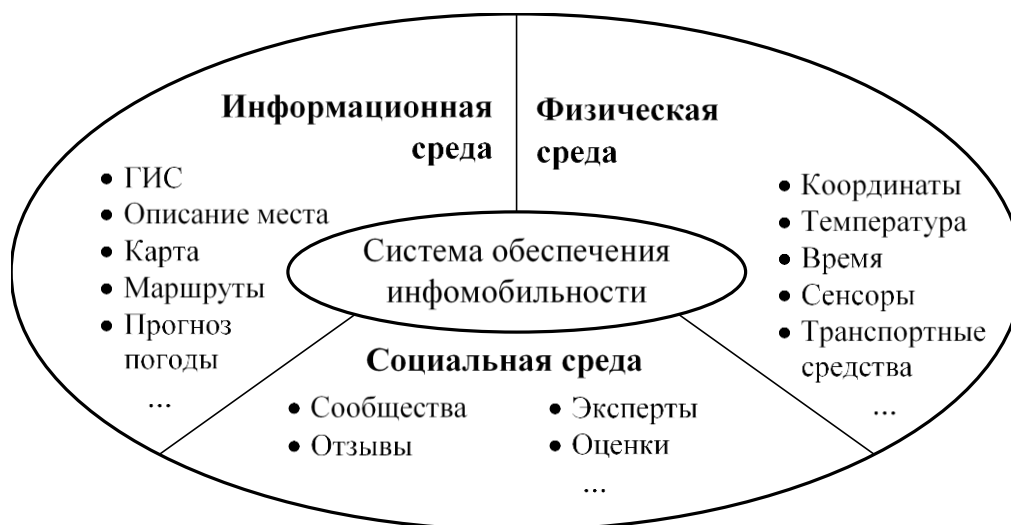


Рисунок 3 — Информационные ресурсы в СОИМ

Вторая категория информационных ресурсов — ресурсы информационной среды. Данная категория объединяет в себе информацию, полученную в результате обработки данных, собранных из других категорий ресурсов, и расширенную информационными составляющими, необходимыми для корректной работы вычислительных систем. В качестве примеров результатов обработки можно привести картографические данные, описания объектов, социальные связи и т.п. Примером информационной составляющей являются идентификаторы, присвоенные пользователям и объектам, ссылки на ресурсы и т.п.

Третья категория — ресурсы социальной среды, предоставляющие информацию о взаимоотношениях между людьми и информацию, полученную от людей и их взаимодействия. Информацией о взаимоотношениях могут быть социальные связи, карьерное положение, информация об образовании и т.п. В ходе взаимодействия пользователей между собой и с системой могут быть получены отзывы о различных объектах, их оценки, экспертная информация и т.п. Данный тип информации вводится пользователем самостоятельно, с помощью интерфейса, предложенного системой.

### Открытые картографические сервисы

Основная часть задач, решаемых системой обеспечения инфомобильности, связана с обработкой географической информации. Источниками данной информации являются географические информационные системы, которые могут также предоставлять сервисы, для ее обработки. Для применения ГИС в системе обеспечения инфомобильности она должна удовлетворять следующим требованиям:

- Максимальный охват и точность данных;
- Поддержка геокодирования (поиска объектов по заданным координатам) и обратного геокодирования (поиска координат по заданным объектам);
- Возможность отображения карты и дополнительной информации поверх нее;
- Поддержка планирования маршрутов и навигации с использованием различных видов транспорта;
- Быстрое построение маршрутов;
- Отсутствие ограничений на использование;

В настоящее время поиск ГИС, удовлетворяющей представленным требованиям не является серьезной проблемой. Наибольший вклад в развитие ГИС как сервисов в сети Интернет осуществлен мировыми корпорациями, занимающимися поисковыми системами, такими как Google, Microsoft, Яндекс. Основные функции, предоставляемые в данных ГИС основаны на отображении карты и функциях, связанных с поиском объектов на карте. Часто список функций дополнен планированием маршрутов для различных видов транспорта, что позволяет их использовать в системе обеспечения инфомобильности. Например, при помощи сервисов, предоставленных компаниями Яндекс и Google возможно осуществлять поиск интермодальных маршрутов как внутри городов, так и между городами. Оба сервиса могут строить интермодальный маршрут внутри города по всем имеющимся в городе видам транспорта, без учета расписания. Между городами обеспечивается построение маршрута только путем железнодорожного и авиасообщения, но с учетом расписания. В то же время, Яндекс предлагает отдельный сервис Яндекс.Расписания [30], позволяющий получить все возможные варианты передвижения между городами с учетом расписания и ориентировочной стоимости поездки.

Кроме имеющихся сервисов возможно построение собственного сервиса на основе открытого картографического проекта OpenStreetMap (OSM) [129], предоставляющего возможность свободного использования карты мира, созданной интернет-сообществом. Данные карты, помимо основной картографической информации об улицах, содержат маршруты общественного транспорта внутри городов и в пригородах. Информация о междугородних перевозках может быть получена из других источников, в том числе и с помощью сервиса Яндекс.Расписания.

В таблице 2 представлено сравнение существующих картографических сервисов по основным параметрам, характерным для инфомобильных систем.

Таблица 2 — Сравнение основных картографических сервисов

	<b>Карты Google</b>	<b>Here</b>	<b>Яндекс</b>	<b>OpenStreetMap</b>	
Точки интереса	+	+	Только Россия, страны СНГ и Турция	+	
События на карте	+	+		Содержит только наиболее вероятные заторы	
Мобильность	Android, Windows Phone, iOS				
Прямое и обратное геокодирование	+	+	+	+	
Планирование и отображение пути:				Отдельные приложения и сервисы	
<ul style="list-style-type: none"> <li>• Пешеход</li> <li>• Личный автомобиль</li> <li>• Общественный транспорт</li> <li>• Интермодальные маршруты</li> </ul>	Внутри и между городами	Внутри города	Только в интермодальном маршруте	Внутри и между городами	
		Внутри и между городами			
		Внутри города			
		Внутри города			
Поддержка пользователя:					
<ul style="list-style-type: none"> <li>• Навигация</li> <li>• Оповещения о событиях</li> <li>• Динамическое изменение маршрута</li> </ul>	+	+	+	Зависит от отдельных приложений и сервисов	
	-	-	+		
	+	-	+		
Добавление и редактирование информации пользователями	+	+	+, слой «народная карта»	Вся карта создается и редактируется пользователями	
Дополнительные сервисы	Фотографии, панорамы, бронирование отелей, расписание пригородных поездов, расписание движения общественного транспорта	Фотографии, панорамы, отображение погоды	Фотографии, панорамы, Яндекс.Расписание Яндекс.Транспорт Яндекс.Такси	-	

Продолжение таблицы 2

	<b>Карты Google</b>	<b>Here</b>	<b>Яндекс</b>	<b>OpenStreetMap</b>
Ограничения	<p>Не более 2500 расчетов путей в день (с лицензией — до 100000 путей в день)</p> <p>Допускается использование до 8 узловых точек</p> <p>Длина HTML запроса ограничена 2048 символами</p>	<p>100000 запросов в месяц</p>	<p>Не более 25000 запросов в сутки к геокодеру и не более 25000 запросов к маршрутизатору</p>	<p>Ограничение на количество запросов по визуализации карты. Но разработчики могут настроить свой сервер для визуализации карты и планирования маршрутов.</p>

Сервис Google Maps [85] впервые стал доступен широкой публике в 2005 году, предоставляя сначала только подробные карты Северной Америки. К настоящему времени на картах Google кроме картографической информации, доступной для всего мира, предоставляется доступ к большому числу сервисов, обеспечивающих пользователя дополнительной информацией и повышающих инфомобильность пользователя.

Сервис Here [91] является развитием картографического проекта компании Navitq, выкупленного Nokia в 2007 году. Предоставляет подробные карты более чем для 180 стран мира, а также дополнительные сервисы навигации, поиска точек интереса и отображения расширенной информации о них.

Главный недостаток обоих представленных выше сервисов — неполный географический охват. Хотя общие карты доступны для всего мира, их подробность и список предоставляемых сервисов сильно отличается в зависимости от региона. Так, например, пользователям из России, стран СНГ, восточной Европы и Азии, недоступны многие события на карте (отображаются только заторы), не во всех городах имеется расписание движения транспорта или охвачен не весь имеющийся транспорт. Такая ситуация складывается из-за того, что оба сервиса используют сервисы третьей стороны для получения дополнительной информации, и обработка данных для собственных сервисов происходит в полуавтоматическом режиме.

Российской компанией Яндекс представлен набор отдельных приложений, которые совместно позволяют обеспечить инфомобильность: Яндекс.Карты [29] (отображение карт, точек интереса, геокодирование, построение маршрутов, заторы и события), Яндекс.Навигатор (построение маршрутов, навигация по маршруту), Яндекс.Транспорт (отображение текущего местоположения общественного транспорта, прогноз времени ожидания на остановке), Яндекс.Расписание (расписания всех видов транспортных средств от маршрутных такси до

самолетов), Яндекс.Такси [31] (поиск и заказ такси). Однако географический охват сервисов достаточно мал. В списке регионов, в которых обеспечена полная и надежная работа каждого из сервисов (кроме карт и навигации, доступных по всему миру) значится только Россия, страны СНГ и Турция.

Среди open-source проектов наибольшую известность в области картографических систем имеет проект OpenStreetMap [129] — это Википедия-подобный проект, в котором карта мира составляется усилиями сообщества. Данные о дорогах обычно получаются из треков, записываемых на GPS-приемники и импортируемых в специальные программы, для построения карты. При этом, использование уже имеющейся картографической информации допускается только при условии, что это не противоречит интересам правообладателя. К настоящему моменту он предоставляет в свободном доступе наиболее подробную карту мира, позволяя разработчикам использовать картографические данные для построения собственных сервисов. Разработаны приложения, использующие данные карт OpenStreetMap для поиска достопримечательностей, построения маршрутов, навигации и как источник информации для картографических сервисов.

Помимо сервисов, перечисленных выше, существуют сервисы, предоставляющие ограниченный набор функций. Их можно поделить на два больших класса: навигационные приложения и туристические гиды. Навигационные программы (Навител, СитиГид и др.) имеют в целом однообразный функционал, который, помимо простого построения маршрута, позволяет в реальном времени получать информацию о заторах, различных событиях на дороге и, в случае необходимости, автоматически перестраивать маршрут с учетом текущей ситуации. Туристические гиды (Triposo, TripAdvisor [170], TAIS [24] и т.д.) обеспечивают поддержку туриста дополнительной информацией, включающей в себя туристические карты городов, погоду, маршруты между основными достопримечательностями, описание достопримечательностей, фотографии пользователей, их отзывы и оценки.

Представленные выше сервисы покрывают большую часть потребностей пользователя, связанных с его инфомобильностью. Как видно из таблицы 1, ни один из сервисов не реализует полный набор ожидаемых функций в единой системе.

Наибольшая точность и актуальность картографической информации предоставляется проектом OpenStreetMap. Данные карт обновляются ежедневно, крупные обновления принимаются еженедельно. При организации собственного картографического сервиса на основе OpenStreetMap есть возможность настроить частоту обновлений картографической информации с основного сервиса.

Отображение актуальной информации о заторах, ДТП и ремонтах дорог зависит от провайдеров подобных услуг в регионе. По всему миру подобные услуги оказывает Google, однако объем предоставляемых данных отличаются в зависимости от географического региона.

На территории стран СНГ самая актуальная информация о текущем состоянии дорог предоставляется компанией Яндекс.

### **Совместное использование личного транспорта**

Система обеспечения инфомобильности предполагает включение личного транспорта в инфраструктуру общественного транспорта. Суть совместного использования транспорта заключается в объединении нескольких пользователей, имеющих близкие по координатам точки начала и конца пути, в один маршрут с использованием только одного транспортного средства. Данный шаг позволяет распределить транспортный поток таким образом, чтобы для перевозки того же количества людей использовалось меньшее количество транспортных средств, а расходы на их содержание равномерно распределялись между участниками совместного использования автотранспорта. Использование данного подхода в системе обеспечения инфомобильности позволяет предоставить пользователям дополнительный способ передвижения между заданными точками, с возможностью выбора попутчиков и приемлемой стоимостью поездки.

Идеи совместного использования автотранспорта (также известного, как карпул, от английского *carpool*) [57] существуют с начала 20 века. Первым успешным массовым проявлением карпула считается 1914 год, когда в результате Первой Мировой войны экономика США впала в рецессию и некоторые предприимчивые владельцы автомобилей в Лос-Анджелесе, желая слегка сэкономить и, возможно, подзаработать начали подбирать пассажиров на своем пути. При этом в качестве вознаграждения водителю по всему городу закрепилась плата в пять центов (в английском *jitney*), отчего этот период получил название "Jitney Craze" (пятицентовое сумасшествие) [144].

Следующим важным периодом в истории совместного использования автотранспорта, а также периодом, наиболее часто упоминающимся, как первое проявление карпула в современном его понимании, стал период Второй Мировой войны. В противовес предыдущему периоду, американское правительство стало серьезно поощрять карпул, в том числе и денежными дотациями, видя в этом способ сберечь ресурсы (в основном топливные) для нужд войны. Подобное поощрение со стороны правительства было уникальным и серьезно способствовало совместному использованию автотранспорта [144].

Третьим периодом повышенного интереса к карпулу является нефтяной кризис 1973 года, который был вызван тем, что основные арабские экспортеры нефти отказались поставлять нефть странам, поддерживающим Израиль в ходе конфликта с Сирией и Египтом и кризис 1979 года, связанный с революцией в Иране, что опять вызвало сокращение поставок нефти. В течение обоих кризисов цены на топливо существенно возрастали и, в целях экономии, многие стали

использовать автотранспорт совместно, оплачивая топливо и содержание автомобиля общими силами [144].

В наши дни также наблюдается серьезный рост цен на топливо и существенное увеличение количества автомобилей на улицах городов, что способствует возрождению и активному продвижению идеи совместного использования автотранспорта. Этому способствует также и развитие сети Internet и мобильных средств связи, благодаря которым даже люди, не знакомые друг с другом ранее, могут быстро найти друг друга и договориться о деталях поездки.

В результате анализа существующих на сегодняшний день информационных систем, автоматизирующих поиск попутчиков для совместного использования автотранспорта были выделены следующие основные группы систем [8]:

**Поиск через форумы и сообщества.** Общий принцип работы подобной схемы заключается в следующем: пользователь, желающий найти попутчика (водителя или пассажира), размещает на специализированном сайте или форуме объявление о поездке, содержащее краткую информацию о пользователе и предполагаемый маршрут, с указанием начальной и конечной точек. При поиске попутчиков учитываются дополнительные условия, такие как плата за проезд, время поездки и др. Среди примеров таких ресурсов можно выделить:

- «Довезу!ру» [5]. Веб-ресурс, предоставляющий услуги поиска попутчиков на территории более чем 100 городов стран СНГ. Позволяет искать пассажиров, водителей, попутчиков для поездки в общественном транспорте. Также оказываются услуги по поиску ближайших такси и поиска транспорта для перевозки грузов.
- eRideShare.com [71]. Ресурс предоставляет услуги поиска попутчиков по всему миру, но особое развитие получил в США, Австралии и некоторых европейских государствах. Осуществляет автоматический подбор попутчиков для поездок внутри города, и между городами.
- PickupPal [134]. Как и предыдущий сервис, позволяет искать попутчиков не только в черте города, но и при поездках между городами. Наибольшее распространение получил в США, Канаде и Великобритании.
- Zimride [180]. Сервис предоставляет услуги поиска попутчиков в основном для университетских и корпоративных сетей. Охватывает университеты и корпорации в Сан-Франциско, Лос-Анджелесе, Нью-Йорке, Вашингтоне и Озере Тахо.
- RideshareOnline [145]. Кроме простого поиска попутчиков, сервис предлагает программу школьных автобусов. На настоящий момент сервис доступен для жителей Вашингтона и Индианы.



- rideshare.511.org [143]. Как и предыдущий сервис, поддерживает поиск попутчиков на разных видах транспорта и программу школьных автобусов, доступен на всей территории США.
- CarJungle [54]. Система поиска попутчиков, как в черте города, так и между городами. Охватывает всю Российскую Федерацию.
- BlaBlaCar [48]. Позволяет осуществлять поиск попутчиков для путешествий между городами. Действует на территории 19 европейских стран, в число которых входит также Российская Федерация.

**Поиск через закрытые веб-сайты.** Данная категория сервисов характеризуется обязательным наличием приглашения для участия в сообществе. Например, сервис Zimride, наравне с общедоступным интерфейсом имеет приватный, используемый различными организациями для осуществления перевозки работников к месту работы силами коллег.

**Использование специализированного программного обеспечения для поиска.** Для мобильных устройств разработаны приложения, позволяющие осуществлять поиск попутчиков по заданным параметрам. Примерами данной категории могут служить сервисы PickupPal, который кроме специализированного сайта имеет приложение для мобильных устройств, Avego [43] и BlaBlaCar.

**Поиск через агентов** Примером такого способа поиска попутчиков могут служить сервисы Uber [171], Gett [82], «Яндекс.Такси» [31], имеющие приложения для мобильных устройств. Главное отличие от предыдущей категории заключается в том, что эти сервисы предоставляют поиск профессиональных лицензированных водителей, не привязанных к какому-либо таксопарку, что позволяет снизить стоимость поездки за счет отсутствия необходимости содержания диспетчерского центра. Водители приезжают точно по указанному на карте адресу в назначенное время.

### **Виды открытых сервисов, предоставляющих информацию в зависимости от местоположения**

К настоящему времени в сети Интернет доступно значительное количество сервисов, предоставляющих информацию в зависимости от заданного местоположения (англ. LBS — location-based services). Многие из них являются открытыми, предоставляя доступ к информации без ограничения авторского права. С учетом требований, сформулированных в 1.3, в системе обеспечения инфомобильности должны использоваться следующие типы сервисов:

- 1) Поиск погоды. Основным назначением данного типа сервисов является предоставление текущей погоды по заданным координатам. Дополнительно может предоставляться прогноз на 1-4 дня, либо история наблюдений за погодой в

произвольное время, начиная с момента запуска сервиса. Как правило данный тип сервисов является глобальным, то есть предоставляет погоду по всему земному шару. Примерами сервисов, предоставляющих погоду являются GISMeteo [83], WorldWeatherOnline [177] и другие.

- 2) Поиск достопримечательностей. Данный тип сервисов позволяет найти достопримечательности, расположенные в указанном радиусе от текущего местоположения пользователя. Достопримечательности могут быть классифицированы и предоставляться с учетом предпочтений пользователя. Данный тип сервисов может быть локальным или глобальным, Локальные разрабатываются для отдельно взятого географического региона, и представляют из себя отдельное приложение для мобильного устройства, например, St Petersburg Guide Monument [164] Глобальные, позволяют осуществлять поиск достопримечательностей по всему земному шару, например, Wikipedia Nearby [46; 175], Foursquare [76], GeoNames [81] и т.д.
- 3) Поиск информации по местоположению. В отличие от сервисов, осуществляющих поиск только по достопримечательностям, данный тип сервисов позволяет осуществлять поиск по всем видам точек интереса (POI – Points of Interest). Кроме поиска точек на карте, осуществляется поиск дополнительной информации о них, например, описание достопримечательности, время работы организаций, отзывы пользователей и т.д. Примерами данного типа сервисов могут служить уже упомянутые выше Wikipedia Nearby, Foursquare, а также TripAdvisor [170].
- 4) Поиск фотографий по местоположению. Данный тип сервисов позволяет осуществлять поиск изображений, ассоциированных с определенными географическими координатами, многие из которых также снабжаются тегами для удобства категоризации. Благодаря наличию координат и тегов становится возможным отбор изображений, связанных с указанным объектом на карте. Примерами подобных сервисов могут служить Flickr, Instagram, Google Photos и другие.
- 5) Поиск информации о дорожных событиях. С помощью сервисов данного типа можно осуществлять сбор и анализ информации о событиях, происходящих на дороге. К ним можно отнести заторы, дорожно-транспортные происшествия, ремонты дорог и другие. Как правило, подобная информация размещается самими пользователями в режиме реального времени с помощью мобильных устройств. Среди сервисов, осуществляющих сбор и предоставление подобного рода информации можно выделить «Яндекс.Пробки», TomTom Live Traffic [169] и другие.

б) Поиск информации о расположении транспортных средств на маршрутах. Данный тип сервисов носит преимущественно локальный характер, за исключением авиа и железнодорожных перевозок. Благодаря развитию систем глобального позиционирования и оснащению маршрутных транспортных средств отслеживающими устройствами стало возможным в любой момент времени указать их текущее местоположение заинтересованным лицам. Таким образом, с помощью сервисов данного типа в городах возможна оценка времени ожидания транспортного средства на маршруте. Примерами данного типа сервисов являются: на локальном уровне — Портал общественного транспорта г. Санкт-Петербурга [17], на глобальном — FlightAware [111] для отслеживания текущего положения самолетов.

## 1.5 Выводы по главе 1

- 1) К настоящему времени для повышения мобильности населения во многих странах разработаны информационные транспортные системы, основная цель которых — обеспечение пользователей информацией о текущем состоянии транспортной сети и предложение вариантов передвижения между заданными точками с помощью мобильных устройств. Такие системы называются системами обеспечения инфомобильности. Дополнительно в них может выводиться информация о точках интереса, погоде в текущем расположении, фотографии пользователей, а также могут предоставляться различные сервисы, например, бронирование отелей, покупка билетов на мероприятие в ближайшем концертном зале и т.д. Подобные системы чаще всего строятся с применением концепции повсеместных вычислений, которые на основе контекстной информации каждого пользователя, в проактивном режиме предлагают информацию и сервисы, которые с наибольшей вероятностью будут полезны пользователю в текущий момент.
- 2) Существующие ИТС реализуют большую часть функций, требуемых для обеспечения инфомобильности. Однако их анализ показал, что качество предоставляемой информации может быть повышено за счет объединения в одной системе функций, которыми обладают все рассмотренные системы. В ходе анализа работ, посвященных построению систем, обеспечивающих инфомобильность, был сформулирован список требований, которые следует учитывать при разработке подобных систем. Этими требованиями являются:
  - 1) Использование открытых данных;
  - 2) Использование сервисов при построении системы;
  - 3) Поддержка планирования мультимодальных маршрутов и совместного использования личного транспорта;
  - 4) Учет предпочтений пользователя;

- 5) Работа в режиме реального времени;
  - 6) Учет текущей ситуации;
  - 7) Конфиденциальность информации о пользователе.
- 3) Анализ существующих подходов к построению систем обеспечения инфомобильности показал, что не все системы полностью удовлетворяют сформулированным выше требованиям. Дополнительно учитывая динамику роста количества транспорта на дорогах, разработка системы, обеспечивающей инфомобильность населения с учетом представленных требований является актуальной задачей, решение которой позволит повысить удобство передвижения по региону.
  - 4) При построении подобной системы централизованная архитектура, в которой все функции выполняются на одном центральном вычислительном кластере, является неприемлемым решением в силу того, что каждая из функций требует обработки достаточно большого объема данных и существенных затрат вычислительных ресурсов. Кроме того, централизованная архитектура существенно затрудняет масштабируемость и обеспечение поддержки большого количества пользователей.
  - 5) Распределенная архитектура, предлагаемая авторами всех работ по инфомобильным системам, позволяет реализовать каждую функцию системы как отдельный модуль. Подобный подход делает возможным поддержку большого количества разнообразных устройств, без необходимости существенной переработки системы. Добавление дополнительной функциональности осуществляется запуском нового программного модуля, использующего результаты работы имеющихся в системе модулей и добавляющего к ним свои результаты работы.
  - 6) В качестве источников информации при построении сервисов, выступают материальные и нематериальные объекты из окружения, в котором можно выделить следующие среды, согласно характеру предоставляемой информации: физическую, информационную и социальную. Информация из сред обрабатывается соответствующими сервисами и распределяется в системе для решения текущей задачи по обеспечению инфомобильности. При этом в качестве основных источников информации предлагается использовать источники открытых данных и открытые информационные сервисы.

В следующей главе предложены основные модели, позволяющие построить систему обеспечения инфомобильности, соответствующую сформулированным требованиям, из которых выведены основные принципы системы. Предлагается модель организации сервисов с использованием сервис-ориентированного подхода, сценарная, концептуальная и онтологическая модели системы.

## Глава 2. Подход и модели построения сервис-ориентированных систем обеспечения инфомобильности

### 2.1 Подход к построению систем обеспечения инфомобильности

#### Основные принципы построения систем обеспечения инфомобильности

Опираясь на требования к системе обеспечения инфомобильности, сформулированные в разделе 1.3, можно выделить следующие принципы, лежащие в основе предлагаемого подхода к построению систем обеспечения инфомобильности [22]:

- *Открытость данных.* Основные ресурсы, предоставляющие данные для системы обеспечения инфомобильности, не должны быть связаны лицензионными ограничениями на их использование. Как правило, открытые данные распространяются в форматах, пригодных для машинной обработки, что значительно упрощает их использование. Результаты работы системы, как и исходные коды ее основных частей также должны быть общедоступны.
- *Использование онтологий.* Открытые данные могут быть представлены в различных форматах: от сырых данных в виде списка или таблицы, до данных, структурированных с помощью языка разметки XML. Для использования в системе обеспечения инфомобильности все данные должны быть приведены к структурированному, машиночитаемому формату, также понятному человеку. Таким форматом является представление знаний с помощью онтологии.
- *Распределенность архитектуры.* Данный принцип выводится из требования, определяющего использование большого количества разнообразных ресурсов, предоставляющих открытые данные, каждый из которых может быть представлен соответствующим сервисом. В подобном случае, использование централизованной архитектуры накладывает значительные ограничения на количество поддерживаемых сервисов и их производительность. Также невозможно быстрое масштабирование системы. Распределенная архитектура позволяет подключать к системе произвольное количество сервисов, обеспечивая их взаимодействие и возможность простого добавления и замены сервисов. Также значительно упрощается задача масштабирования системы.
- *Ориентированность на пользователя.* Так как основной целью системы является обеспечение инфомобильности в сфере пассажирских перевозок, пользователь должен быть обеспечен удобным инструментом для работы с системой. Удобство, помимо простого, интуитивно-понятного интерфейса, подразумевает персонализацию работы

системы для предоставления информации, интересной конкретному пользователю, выработку рекомендаций на основе отзывов других пользователей и фоновую проактивную поддержку пользователя при планировании маршрута и следовании по нему.

- *Использование контекстной информации.* Для персонализации результатов работы системы, выработки рекомендаций, прогнозирования развития ситуации для проактивной поддержки пользователя и адаптации системы к внутренним изменениям, система должна использовать имеющуюся информацию о пользователе и его окружении (контекст). Такой информацией является: дорожная ситуация (пробки, ДТП, ремонты дорог, погода и др.), информация о пользователе (имя, возраст, социальное положение и др.), информация о транспортном средстве (марка, текущее состояние узлов, уровень топлива и др.).
- *Самоконтекстуализация сервисов.* Ввиду поддержки большого количества разнообразных сервисов и распределенности архитектуры, набор, количество и свойства сервисов могут меняться с течением времени. Также сервисы влияют на окружение, в котором они работают. В связи с этим сервисы, входящие в систему обеспечения инфомобильности должны быть способны к самоконтекстуализации — адаптации поведения к подобным изменениям в зависимости от контекста.
- *Работа в реальном времени.* Для решения задачи обеспечения инфомобильности в сфере пассажирских перевозок система должна обладать актуальными данными и предоставлять результаты обработки этих данных в режиме реального времени. Поддержание контекста пользователя и информации о текущей дорожной ситуации в актуальном состоянии позволит максимально быстро выявлять текущие потребности пользователя и вызывать соответствующие им реакции системы.
- *Мультимодальность маршрутов.* Одна из основных целей системы обеспечения инфомобильности – повышение мобильности населения при перемещении по региону, достигаемое за счет более полного использования имеющейся транспортной инфраструктуры. Система предоставляет как альтернативные маршруты, так и обеспечивает возможность комбинации различных видов транспорта в одном маршруте, с учетом расписания движения.
- *Конфиденциальность информации о пользователях.* Персонализация влечет за собой необходимость сбора и анализа большого количества информации о пользователе. Система должна обеспечивать недоступность этой информации сторонним лицам, для чего необходимо использование специализированного сервиса для контроля и разграничения доступа. Сбор информации, не связанной с персонализацией

результатов работы системы, должен проходить исключительно в обезличенном виде для невозможности идентификации и последующей компрометации источника.

### **Описание подхода к построению системы обеспечения инфомобильности**

Подход, предлагаемый к использованию при построении системы обеспечения основывается на сервис-ориентированной архитектуре, что предполагает представление открытых ресурсов с помощью сервисов, взаимодействующих друг с другом в виртуальном пространстве взаимодействия. Сервис-ориентированная архитектура (от англ. Service-oriented architecture — SOA) является развитием следующих распределенных архитектур: основанной на использовании компонент (Component Based Architecture), объектно-ориентированного (Object Oriented) подхода, основанного на использовании интерфейсов (Interface Based Design), и распределенных систем, таких как DCOM, CORBA, J2EE. Выбор сервис-ориентированной архитектуры обеспечен следующими ее преимуществами [72; 73; 100; 137; 140; 173]:

- Автономность сервисов. Сервисы, работающие в рамках одной системы, могут быть разработаны различными компаниями независимо друг от друга;
- Абстракция сервисов. Формализация описания сервисов и их интерфейсов в сервис-ориентированной архитектуре осуществляется лишь настолько, насколько это необходимо для их эффективного использования. Оно может не включать информацию об особенностях реализации сервиса, технологий, используемых в сервисе и т.п.;
- Стандартизация. Сервис-ориентированная архитектура, являясь по сути подвидом распределенной архитектуры, для своего функционирования требует обеспечение доступа сервисов друг к другу с точки зрения возможности взаимного обнаружения и последующего использования. Таким образом, при разработке сервисов подразумевается следование определенным стандартам, например, использование типовых форматов передачи данных, общей терминологии и т.п.;
- Возможность повторного использования сервисов. Подразумевает использование одного и того же сервиса в нескольких сценариях или задачах. Сервисы, являясь автономными и абстрактными, и использующими утвержденные стандарты, могут быть использованы различными потребителями информации без необходимости переработки сервиса под нужды новой задачи;
- Композиция. Несколько сервисов могут быть виртуально объединены в комплексный мета-сервис. Существуют различные механизмы организации сервисов в мета-сервисе, такие, как оркестрирование (централизованная организация) и хореография (децентрализованная организация);

- Обнаруживаемость. За счет механизмов абстракции и стандартизации сервис-ориентированная архитектура позволяет осуществлять поиск сервиса для решения конкретной задачи.

Приведенные преимущества сервис-ориентированной архитектуры позволяют удовлетворить основные требования по организации системы обеспечения инфомобильности, объединяя в себе преимущества всех подходов и архитектур, представленных в обзоре систем обеспечения инфомобильности, проведенном в разделе 1.2. Однако приведенные преимущества также влекут за собой ряд недостатков, которыми являются:

- Большое количество передаваемых служебных данных. Автономность, абстракция и стандартизация сервисов вместе с обеспечением возможности обнаружения приводят к тому, что сервисам необходимо обмениваться большими объемами данных для согласования совместной деятельности, что повышает нагрузку на используемую вычислительную технику;
- Сложное управление и тестирование системы. Большое количество автономных сервисов, представленных различными разработчиками затрудняет их совместное тестирование. Кроме того, разнообразие сервисов затрудняет реализацию крупных сценариев, требующих композиции сервисов;
- Сложность разработки и внедрения. При разработке сервисов нужно уделять особое внимание реализации интерфейсов для взаимодействия сервисов в системе. Сложность тестирования также повышает сложность разработки ввиду того, что невозможно провести быструю отладку сервиса на всех возможных сценариях;
- Сложность обеспечения безопасности информации. Традиционно реализация модели обеспечения безопасности информации требуется только для одного конкретного приложения без рассмотрения его связей с другими, что неприемлемо для распределенных систем ввиду большого количества отдельных сервисов, взаимодействующих между собой, что требует разработку новых моделей обеспечения безопасности информации.

Хотя приведенные недостатки сервис-ориентированной архитектуры могут усложнить разработку системы обеспечения инфомобильности, преимущества такой архитектуры позволяют разработать подход, обладающий большой гибкостью и возможностью расширения.

Кроме того, подход к построению системы обеспечения инфомобильности является контекстно-управляемым. В настоящее время множество исследований посвящено использованию контекста при решении задач, стоящих перед сервисами, что обусловлено распространением так называемых повсеместных вычислений (ubiquitous computing), при



которых вычислительные и информационные технологии включаются в среду обитания. Подобное включение позволяет обеспечить сбор информации о физическом окружении и учесть его при выработке решения задачи для более полного соответствия запросу пользователя. Учет контекста при решении задачи позволяет создавать шаблоны поведения и модели описания контекста для типовых ситуаций (смотри 0).

Общая схема сервис-ориентированного подхода представлена на рисунке 4. Открытые ресурсы подразделяются на три категории, согласно разделу 1.4: ресурсы физической среды, ресурсы информационной среды и ресурсы социальной среды. Каждый из ресурсов может напрямую взаимодействовать с другими ресурсами из своей категории и косвенно — через соответствующие сервисы в пространстве взаимодействия сервисов — с ресурсами из всех сред, что предполагает разработку сценарной модели для описания подобного взаимодействия (раздел 2.2).

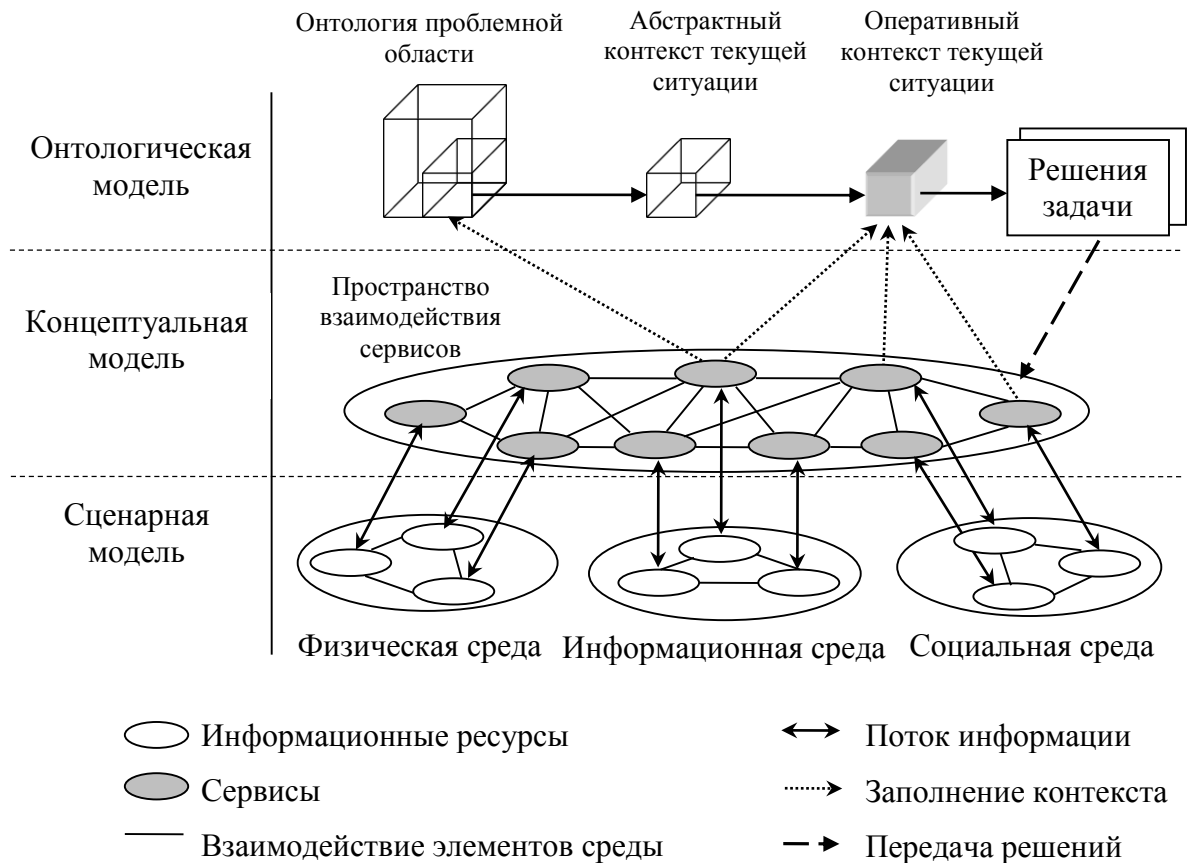


Рисунок 4 — Сервис-ориентированный подход к построению СОИМ

Сервис — основной действующий субъект системы обеспечения инфомобильности. Он может представлять сенсор, транспортное средство, компанию – поставщика услуг и т.п. Сервисы характеризуется такими свойствами как самоконтекстуализация, самоадаптация, автономность, проактивность и способность выполнения некоторых действий.

Пространство взаимодействия сервисов – это виртуальная логическая среда, реализующая принципы сервис-ориентированной архитектуры. В данном пространстве происходит обмен информацией между сервисами, представляющими информационные ресурсы и пользователей системы. Реализация пространства взаимодействия описывается концептуальной моделью (раздел 2.3).

Семантическая интероперабельность сервисов в пространстве взаимодействия обеспечивается представлением знаний помощью онтологии. Преимуществом использования онтологии для представления знаний в системе обеспечения инфомобильности является возможность определения связей между основными понятиями проблемной области в виде пригодном как для машинной обработки, так и понятном для человека. В онтологии каждого сервиса можно выделить модель его абстрактного контекста, определяющую структуру описания текущей ситуации. При наполнении абстрактного контекста информацией из источников формируется оперативный контекст, используемый для выработки решений задачи, стоящей перед системой обеспечения инфомобильности. Таким образом осуществляется учет контекста при выработке решения и управление сервисом, путем изменения контекста. Онтологическая модель сервиса, реализующая данный подход описана в разделе 2.4.

## **2.2 Сценарная модель сервис-ориентированной системы обеспечения инфомобильности**

В результате анализа публикаций, посвященных системам обеспечения инфомобильности, также были определены основные потребности потенциальных пользователей разрабатываемой системы. Проведенный анализ позволил сформулировать типовые пользовательские сценарии для работы с системой, объединенные в общую сценарную модель СОИМ (рисунок 5). Отдельные сценарии могут быть объединены в три крупные группы:

- 1) *Планирование маршрута.* Система производит планирование маршрута, при необходимости совмещая несколько видов транспорта в ходе следования по маршруту, с учетом расписания движения общественного транспорта, и обеспечивая возможность совместных поездок с использованием личного автотранспорта. Сценарий позволяет сократить время ожидания в точках пересадки и произвести планирование маршрута в реальном времени.
- 2) *Поиск объектов.* Система обеспечивает информационную поддержку пользователя при планировании маршрута между объектами на местности. Кроме маршрута пользователю предоставляется рекомендации о ближайших точках интереса в зависимости от контекста, ранжированные на основе отзывов других пользователей, с использованием коллаборативной фильтрации.

- 3) *Визуализация контекста*. На устройстве пользователя отображается информация о текущей ситуации вокруг него: текущая погода, расположение транспортных средств на маршрутах общественного транспорта, заторы и другие дорожные события.

Каждый сценарий для своей работы требует использование нескольких сервисов, позволяющих разделить крупную задачу на составные части, выполняемые параллельно. Требуемые сервисы и их взаимодействие будет рассмотрено в соответствующих разделах.

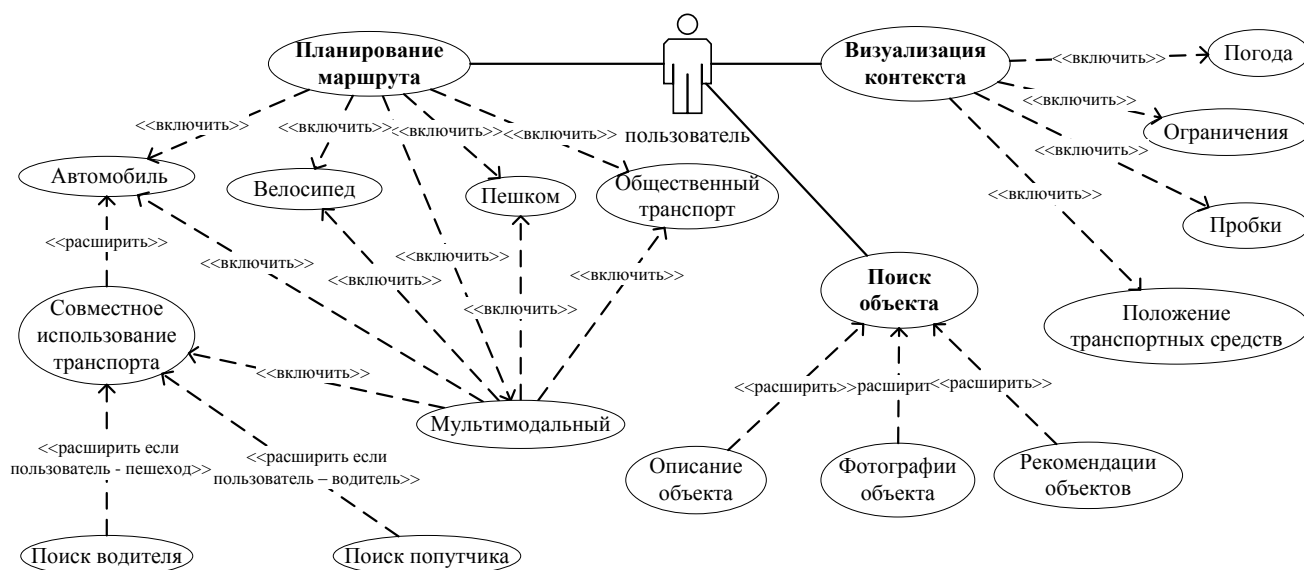


Рисунок 5 — UML-диаграмма основных сценариев использования СОИМ

### Сценарий планирования маршрутов

Данный сценарий позволяет пользователям планировать варианты маршрутов между заданными точками. Маршрут может планироваться для всех имеющихся в регионе видов транспорта, среди которых можно выделить автомобильный, велосипедный, пешеходный и общественный транспорт (автобус, троллейбус, легкорельсный трамвай, метрополитен). Для этого осуществляется обращение к сервисам, отвечающим за каждый из видов транспортных средств. Для автомобильного транспорта возможно обращение к таксомоторным компаниям для предоставления услуг такси, а также планирование совместных поездок с использованием личного транспорта. Целесообразность обеспечения возможности планирования совместных поездок с использованием личного транспорта подтверждается многочисленными исследованиями, связанными с преимуществами подобного вида передвижения, среди которых можно выделить: снижение количества транспорта на дорогах (до 11% по исследованию университета Дели [65]), сокращение общего пробега автомобилей (на 9-27 % согласно исследованию MIT [144]) распределение стоимости поездки и содержания автомобиля между попутчиками. Интерес к подобному способу передвижения подтверждается также опросом,

проведенным в сети Интернет совместно СПИИРАН и университетом Оулу в 2013 году [160], в ходе которого более 65% респондентов согласились на совместные поездки с использованием личного транспорта при дополнительном условии, что водители и/или попутчики будут им знакомы.

Порядок обращения к сервисам показан на рисунке 6. Пользователь, также представленный сервисом, через пространство взаимодействия размещает запрос на поиск маршрута, который распространяется между сервисами, осуществляющими планирование для определенного вида транспорта. Также запрос получает сервис поиска мультимодальных маршрутов использующий композицию сервисов, планирующих маршрут по отдельным видам транспорта для планирования маршрута с использованием комбинации нескольких видов транспорта в ходе поездки.

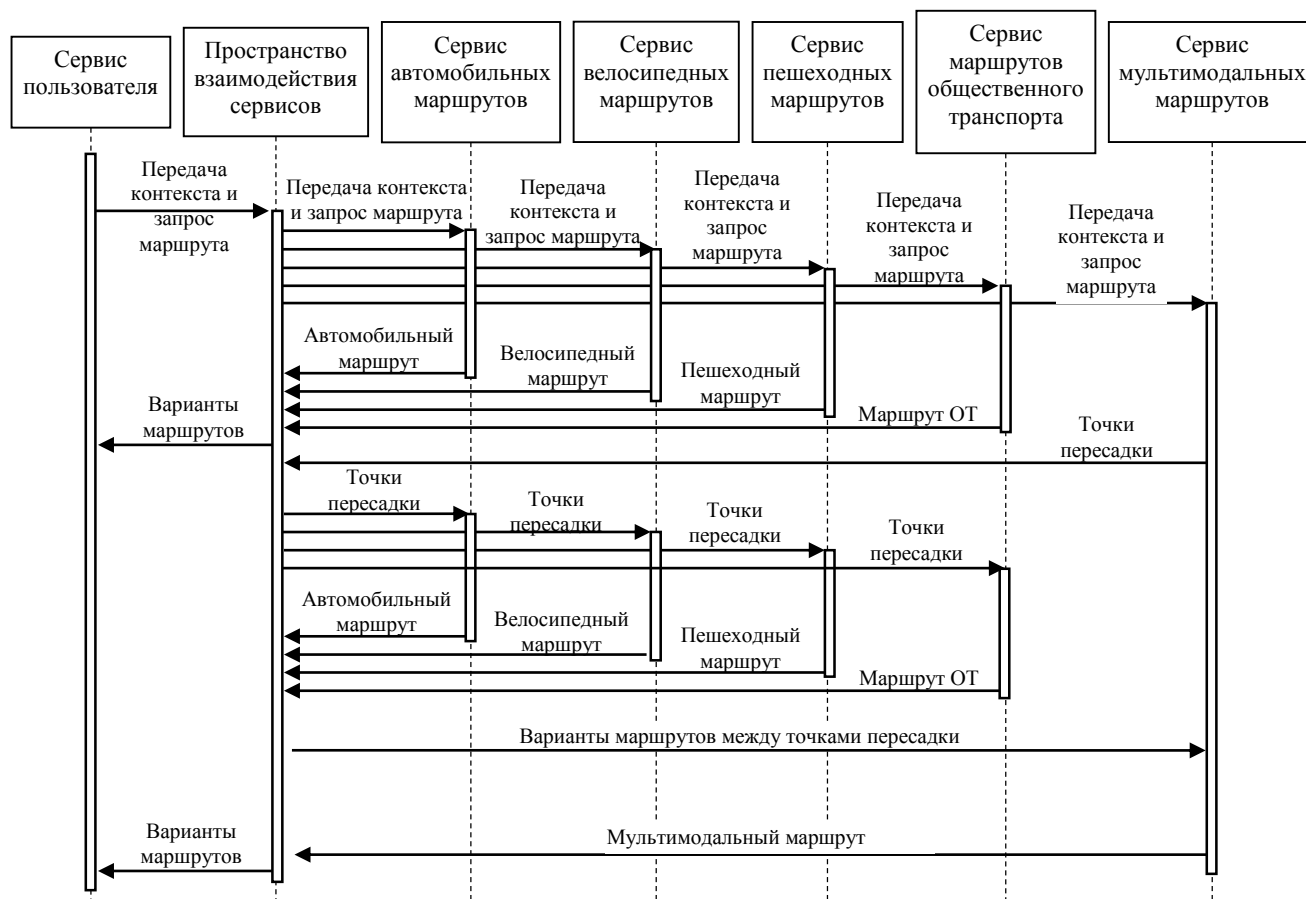


Рисунок 6 — Сценарий планирования маршрутов

Сервисы, осуществляют планирование пути, учитывая особенности транспортных средств. Так, сервис велосипедных маршрутов осуществляет планирование поездки с использованием велосипеда и при планировании отдает приоритет имеющимся в регионе велосипедным дорожкам. Сервис маршрутов общественного транспорта осуществляет планирование только для сети общественного транспорта беря за основу его остановки и

имеющиеся маршруты. Сервис автомобильных маршрутов кроме планирования поездки на личном автомобиле позволяет спланировать маршрут с обращением к услугам такси или с осуществлением совместной поездки на личном транспорте пользователей системы обеспечения инфомобильности.

Сервис мультимодальных маршрутов позволяет планировать маршрут, использующий несколько видов транспорта в ходе поездки. Данная функциональность осуществляется за счет формирования запросов к другим сервисам планирования маршрутов и комбинированием ответов этих сервисов для предоставления целостного маршрута. При этом учитываются особенности движения используемых транспортных средств, например, расписание движения общественного транспорта, стоимость поездки и т.п. Учет данных особенностей позволяет произвести оптимизацию маршрута по заранее определенному критерию, например, снизить стоимость поездки или найти максимально быстрый маршрут.

### **Сценарий поиска объектов на карте**

Целью системы обеспечения инфомобильности является не только планирование маршрута, но и поиск и предоставление информации об объектах на местности. Подобной информацией могут являться:

- Сами объекты. Первоначально сервисы системы должны найти объекты, соответствующие интересам пользователя и его контексту. Такими объектами могут быть различные виды достопримечательностей, пункты питания, отдыха, различные магазины и организации. При этом дополнительно может осуществляться сортировка объектов в зависимости от рейтинга, вычисленного на основе отзывов пользователей системы об этих объектах, с использованием методов коллаборативной фильтрации.
- Описание объекта. Для каждого из найденных объектов осуществляется поиск соответствующей ему информации. Например, для достопримечательностей такой информацией является историческое описание, для магазинов и организаций — время работы и т.д. Поиск информации об объекте осуществляется в источниках открытых данных, таких как *Википедия*, *Викиматия*, и т. п.
- Фотографии объекта. Для упрощения ориентирования на местности пользователю предоставляются фотографии как самого объекта, так и близлежащих мест, позволяющие быстро найти объект. Поиск фотографий осуществляется с использованием открытых сервисов, позволяющих размещать фотографии с привязкой к координатам и осуществлять их поиск по заданному местоположению, например, *Flickr*.

Сценарий поиска объектов показан на рисунке 7. Пользователь размещает свои текущие координаты в пространстве взаимодействия, формируя, тем самым, запрос на поиск объектов, расположенных рядом. Сервис поиска объектов принимает данный запрос и, осуществив поиск по подключенным к нему открытым геоинформационным системам, составляет список объектов, упорядоченный по возрастанию расстояния от пользователя. Данный список через пространство взаимодействия передается в сервис информации, и в сервисы выработки рекомендаций и планирования маршрутов. Информация и фотографии, найденные для каждого объекта, записываются в их свойства в пространстве взаимодействия. Рекомендации, сформированные для полученного списка, также размещаются в пространстве взаимодействия и применяются для упорядочивания списка объектов на стороне пользователя по убыванию рейтинга. Для каждого объекта также отображается соответствующий маршрут, по которому можно добраться до объекта из текущего местоположения пользователя.

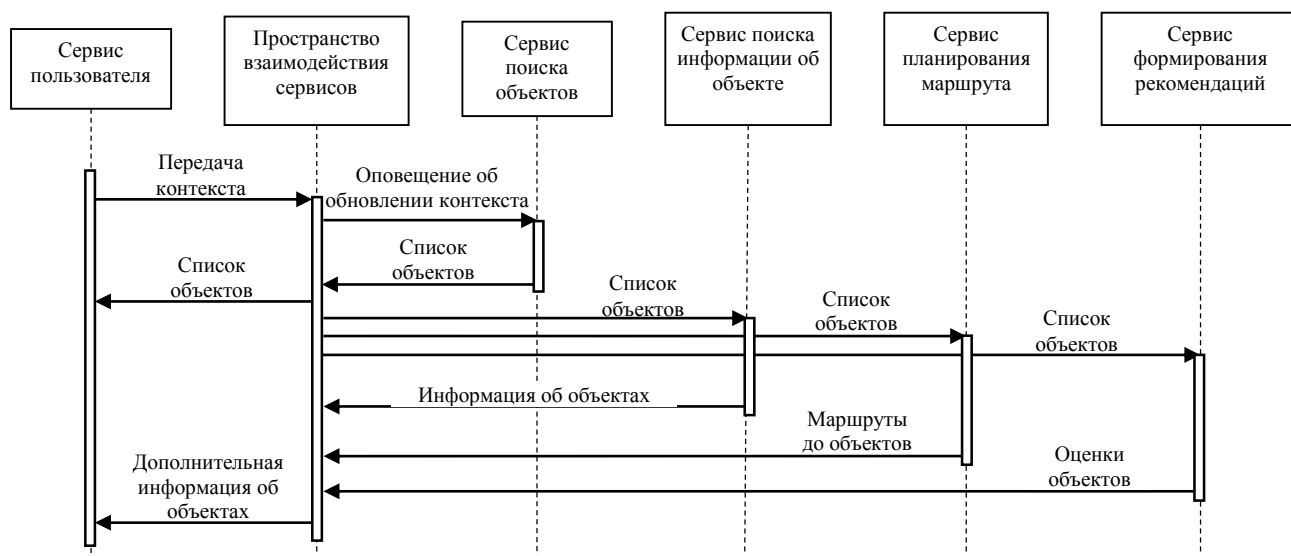


Рисунок 7 — Процесс взаимодействия сервисов в сценарии поиска объекта

### Сценарий визуализации контекста

Сценарий визуализации контекста предполагает сбор информации о текущей ситуации из источников открытых данных и отображение собранной информации на устройстве пользователя. Данная информация может включать в себя:

- Температуру и погодные условия. Источниками информации о температуре могут быть различные сенсоры (термометр, барометр, гигрометр и т.д.), подключенные к системе обеспечения инфомобильности и расположенные рядом с пользователем, а также источники открытых данных в сети Интернет, предоставляющие информацию о погоде в зависимости от заданного местоположения, например, *GISMeteo* [83],

*worldweatheronline.com* [177] и т.д. Кроме текущих погодных условий данные сервисы могут предоставлять прогноз на ближайшее время (1-4 дня).

- Текущее состояние дорожной сети. Под данной категорией подразумевается информация о средней скорости потока, текущая температура дорожного полотна (актуально для автомагистралей) и т. п. Источниками подобной информации могут быть транспортные средства пользователей системы обеспечения инфомобильности, а также открытые сервисы, например, *doroga.tv* [130].
- События на дорогах. Под событиями на дорогах подразумевается информация об авариях, о перекрытых дорогах, используемая при планировании маршрута. Подобная информация размещается в системе обеспечения инфомобильности самими пользователями с привязкой к координатам.
- Расположение транспортных средств на маршрутах общественного транспорта. Развитие навигационных систем позволило обеспечить весь наземный общественный транспорт модулями позиционирования, благодаря которым стало возможно отображение текущего местоположения каждого транспортного средства на карте. Знание текущего местоположения вместе с информацией о текущем состоянии дорожной сети предоставляет возможность спрогнозировать время прибытия транспорта в определенную точку маршрута, что позволяет спланировать маршрут таким образом, чтобы сократить время ожидания на остановках в случае пересадки.

Взаимодействие сервисов для сценария визуализации контекста показано на рисунке 8. Сервисы, отвечающие за сбор контекстной информации, основываясь на текущем местоположении пользователя обновляют контекст в пространстве взаимодействия, откуда он может быть выбран сервисом пользователя и отображен на экране устройства.

## **2.3 Концептуальная модель сервис-ориентированной системы обеспечения инфомобильности**

### **Модель организации системы обеспечения инфомобильности**

Для реализации преимуществ сервис-ориентированной архитектуры и учета большого количества источников информации, пространство взаимодействия сервисов в данной работе реализовано с использованием архитектурной модели «классная доска», которая успешно применяется при решении задач, требующих координации различных процессов или источников знания. Архитектура «классной доски» — это стратегия решения сложных системных задач с привлечением разнородных источников знаний, взаимодействующих через общее информационное поле [14].

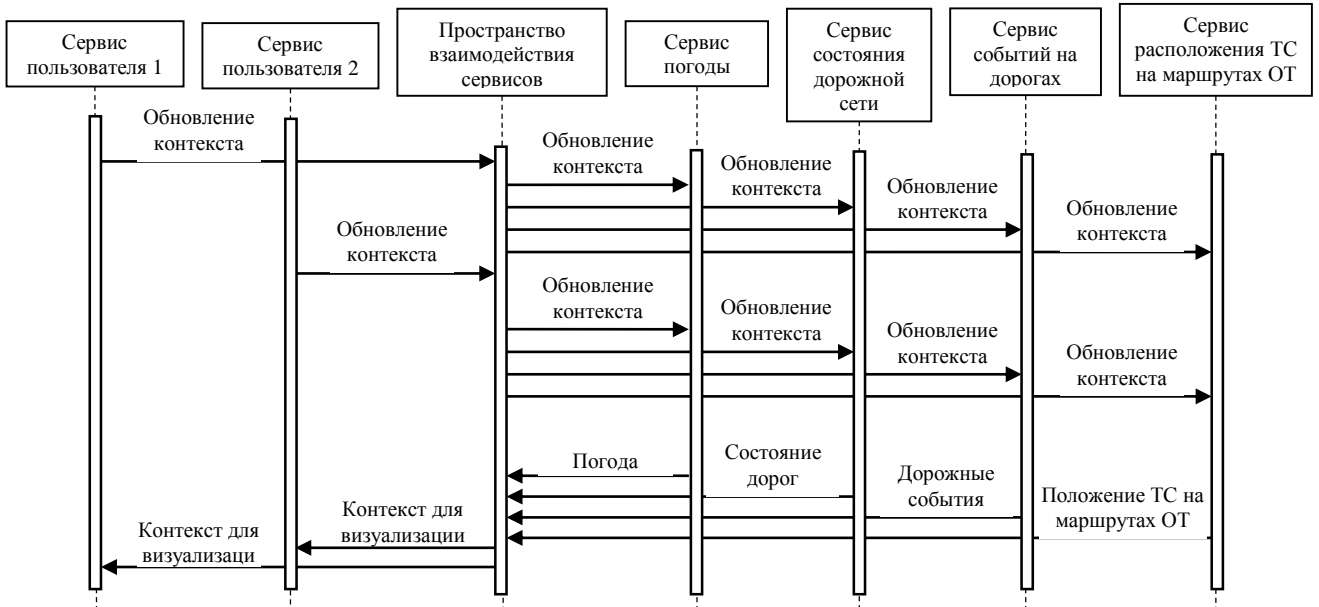


Рисунок 8 — Сценарий визуализации контекста

Использование архитектуры «классная доска» для поддержки сценариев, представленных в разделе 2.2, отображено в концептуальной модели системы обеспечения инфомобильности, представленной на рисунке 9. Она описывает взаимодействие сервисов системы между собой и основные потоки информации в системе.

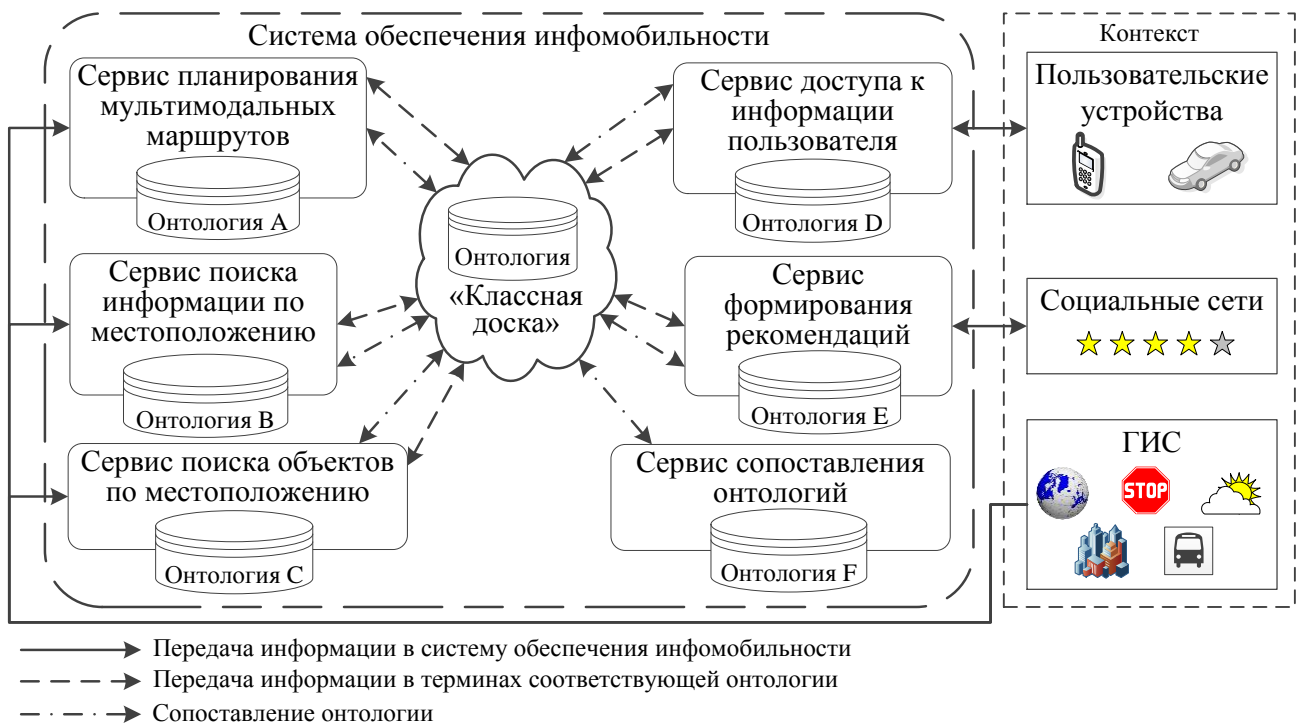


Рисунок 9 — Концептуальная модель системы обеспечения инфомобильности

Архитектура «классной доски» предполагает наличие центральной базы данных, предназначенной для связи независимых асинхронных источников знаний. В качестве источников и обработчиков знаний в системе используются следующие сервисы:



- Сервис планирования мультимодальных маршрутов. Включает в себя планирование маршрута по отдельным видам транспорта и их комбинациям.
- Сервис поиска информации по местоположению. Включает в себя сбор контекстной информации, привязанной к координатам и объектам.
- Сервис поиска объектов по местоположению. Позволяет осуществлять поиск различных объектов (музеи, памятники, организации и т.д.) в зависимости от текущего местоположения пользователя.
- Сервис обеспечения конфиденциальности информации. Данный сервис вводится в систему в силу того, что для обеспечения персональной поддержки требуется сбор данных, которые могут быть использованы третьей стороной с целью компрометации пользователя или установления слежки за ним.
- Сервис формирования рекомендаций. Сервис используется для сбора отзывов об объекте и формирования рекомендаций на основе имеющихся оценок других пользователей.
- Сервис сопоставления онтологий. Данный сервис необходим для обеспечения взаимодействия сервисов разных производителей, позволяя найти схожие понятия в используемых онтологиях, снижая тем самым сложность ввода в систему новых сервисов.

Данные сервисы могут размещать на «классной доске» информацию из разнородных источников, составляющих контекст: пользовательские устройства, оснащенные датчиками, позволяющими получать физические характеристики среды вокруг пользователя; различные ГИС, предоставляющие информацию об объектах, позволяющие производить поиск объектов на местности, планирующие маршруты с использованием различных видов транспорта, предоставляющие погоду по указанному местоположению и т.п.; различные социальные сети и информационные сервисы, позволяющие оценивать различные объекты на местности.

Сервисы являются изолированными друг от друга и оперируют только информацией, доступной в «классной доске» и в соответствующем сервису источнике контекстной информации. После обработки собранной информации, сервис размещает результат в «классной доске» для его использования другими сервисами. Действуя асинхронно, в зависимости от наличия необходимой информации сервисы, таким образом, предоставляют комплексное решение задач, описываемых сценариями, представленными в предыдущем разделе. При этом, в ходе обработки информации сервисы должны быть способны менять свою конфигурацию и поведение в зависимости от текущей ситуации, для обеспечения релевантности решения поставленной задаче. Модификация конфигурации осуществляется путем обеспечения свойства

самоконтекстуализации сервиса, согласно которому сервис может описывать контекст и использовать его для модификации своего поведения.

Чтобы обеспечить обмен информацией между сервисами посредством «классной доски», в данной работе используется представление знаний проблемной области сервисов с использованием онтологии. Онтология — формально представленные на базе концептуализации знания, что предполагает описание множества объектов и понятий, знаний о них и связей между ними [2]. Для предметной области предполагается формальное и декларативное представление, которое включает в себя словарь терминов предметной области, ограничения целостности на термины, логические утверждения, ограничивающие интерпретацию терминов и их отношения друг с другом.

Для каждого сервиса создается онтология, наиболее полно описывающая соответствующую проблемную область. Подобный подход позволяет сосредоточить внимание разработчика сервиса на глубоком анализе конкретной проблемной области с целью максимально точного и емкого ее описания, однако затрудняет использование знаний разработанной онтологии в других сервисах системы обеспечения инфомобильности. В связи с этим требуются разработка и реализация механизма, обеспечивающего поиск соотношений между понятиями, используемыми в онтологиях разных сервисов. Таким механизмом является метод сопоставления онтологий, описанный в следующем разделе.

### **Самоконтекстуализация сервисов в системе обеспечения инфомобильности**

Система обеспечения инфомобильности является динамической системой, в которой сервисы непрерывно создают, изменяют и осуществляют обмен информацией посредством «классной доски». При этом информация, находящаяся в любой момент времени на «классной доске» составляет контекст сервисов и может быть использована для описания текущей ситуации. Очевидно, что для каждой конкретной ситуации описывается определенными уникальными параметрами и требует использования определенных функций сервисов. В связи с этим, при реализации сервисов системы обеспечения инфомобильности необходимо предусмотреть возможность автоматической конфигурации сервиса в зависимости от контекста, описывающего текущую ситуацию.

Для автоматической конфигурации сервиса необходимо, чтобы он мог самостоятельно описывать и использовать контекст, а также изменять свое поведение в зависимости от текущей ситуации [147]. Подобное свойство сервиса называется самоконтекстуализацией. Под поведением сервиса в данном случае понимается способность выполнять определенные действия как внутри себя, так и в своем окружении.

В данном разделе представлен подход, использующий идею самоконтекстуализации для автономной модификации поведения сервисов системы обеспечения инфомобильности согласно контексту текущей ситуации, для принятия ими контекстно-зависимых решений [3]. При построении сервисов используется концептуальная модель, представленная на рисунке 10 [167]. Основные компоненты модели описаны далее.

Абстрактный контекст задает концептуальное описание проблемной области, в которой работает соответствующий сервис. Он представляет собой внутреннюю онтологию сервиса. Абстрактный контекст описывает структуру оперативного контекста сервиса. В зависимости от ситуации абстрактный контекст может быть модифицирован (адаптирован) посредством самоконтекстуализации. Он также описывает терминологию контекста и профиля сервиса.

Оперативный контекст представляет знания о конкретной ситуации. Его структура определяется абстрактным контекстом сервиса, которая наполняется текущими значениями параметров текущей ситуации. Оперативный контекст, отображая текущую ситуацию в окружении, определяет поведение сервиса.

Контекстом называется любая информация, которая может быть использована для описания ситуации в которой находится объект, где в качестве объекта может выступать человек, место и т.п., который считается относящимся к задаче, решаемой сервисом, включая сам сервис и используемые им инструментальные средства [66]. Целью создания контекста является представление только релевантных информации и знаний из всего объема доступных в системе. Релевантность информации и знаний оценивается на основе того, насколько они связаны с рассматриваемой задачей. Структура контекста текущей ситуации описывается с помощью абстрактного контекста. В связи с тем, что в результате воздействия на окружение как других сервисов, так и рассматриваемого сервиса, меняется как само окружение, так и контекст, его представляющий, абстрактный контекст необходимо модифицировать для отображения релевантных параметров контекста. Происходит это через обновление оперативного контекста, который в свою очередь определяет поведение сервиса.

Окружение представляет окружающую среду системы обеспечения инфомобильности, в которой выделяются три типа сред: физическая, информационная и социальная. Элементы окружения представлены сервисами в пространстве взаимодействия (раздел 2.2), частью которого также является рассматриваемый сервис. Окружение способно взаимодействовать с системой посредством сервисов. Окружение влияет на контекст рассматриваемого сервиса. Сервис может воздействовать на окружение, при условии, что он наделен соответствующей функциональностью (например, автобус может перевезти пассажира).

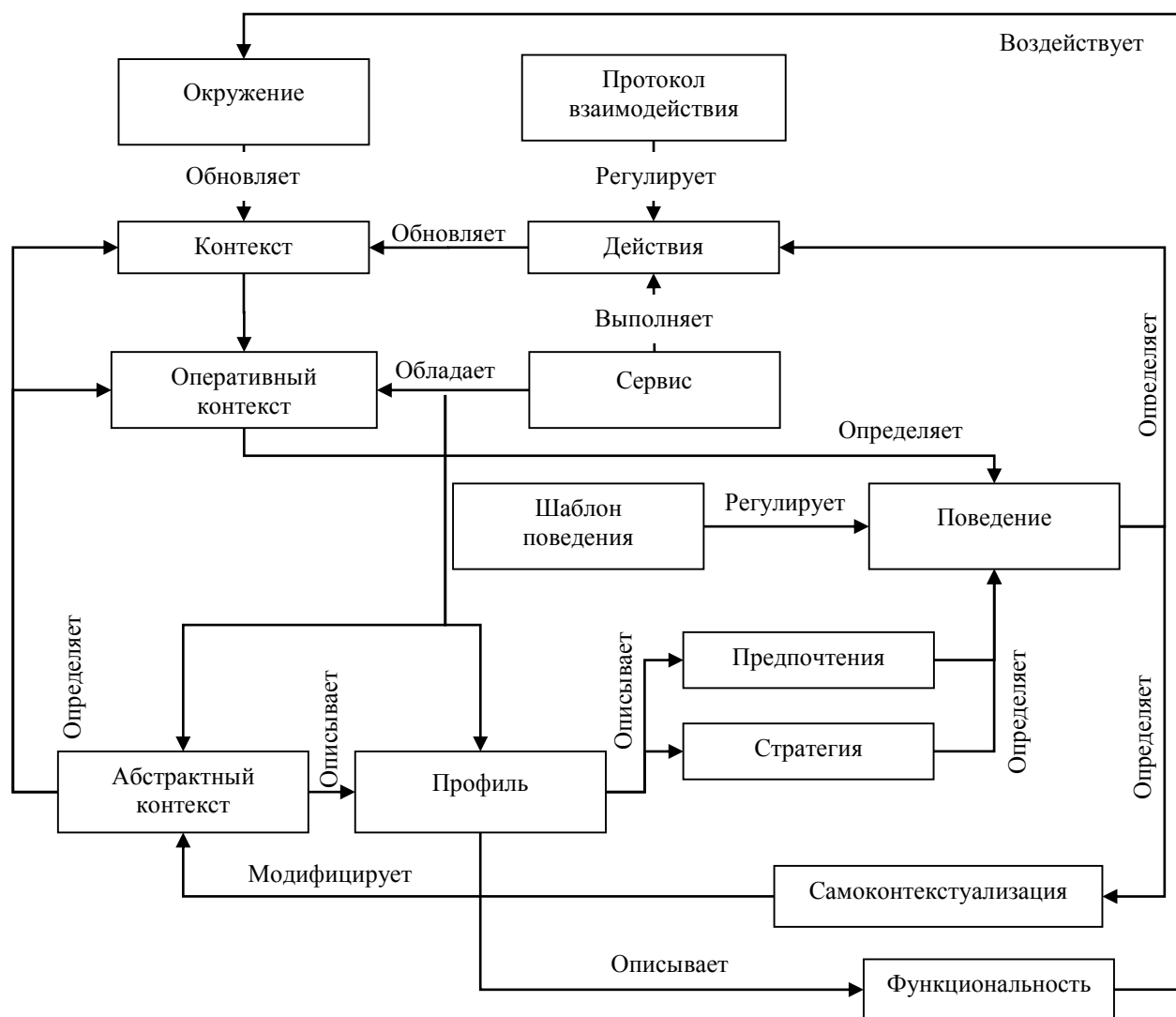


Рисунок 10 — Концептуальная модель самоконтекстуализирующегося сервиса

Под функциональностью сервиса понимается набор функций, которые могут быть выполнены им в своем окружении. Функциональность может быть в некоторой степени изменена посредством самоконтекстуализации, путем включения в абстрактный контекст сервиса новых функций, отвечающих контексту системы. Функциональность описывается профилем сервиса.

В профиле сервиса описывается его функциональность, стратегии и предпочтения в терминах абстрактного контекста. Описание осуществляется таким образом, чтобы другие сервисы системы могли интерпретировать эти характеристики.

Способность сервиса выполнять определенный набор действий, включающих в себя обработку информации, воздействие на окружающую среду и изменение абстрактного контекста называется поведением. Оно определяется предпочтениями и стратегиями, заданными в сервисе, а также определенными шаблонами поведения.

Шаблон поведения — это заранее определенный набор правил и принципов действия, определяющий порядок обработки информации и направляющий поведение сервиса с целью

достижения рационального результата. Шаблоны поведения задаются разработчиками сервиса и предусматривают порядок действий в типовых ситуациях, характерных для него.

Предпочтения сервиса отражают его требования к наличию и/или отсутствию определенных состояний окружения или собственных состояний. Они задаются в абстрактном контексте через профиль сервиса и также могут быть изменены посредством самоконтекстуализации.

Стратегия определяет план выбора шаблонов поведения для удовлетворения требованиям текущей ситуации. Она также описывается абстрактным контекстом в профиле сервиса и может быть модифицирована в процессе самоконтекстуализации.

Сервисы могут также совершать определенные действия, определенные поведением, для осуществления взаимодействия с другими сервисами системы. Эти действия регулируются протоколом взаимодействия.

Протокол взаимодействия определяет базовые правила и порядок действий, по которым осуществляется взаимодействие между сервисами. При следовании этим правилам взаимодействие с другими сервисами будет гарантированно приводить к обмену информацией и определенным результатам совместной работы, в том виде, как это было запланировано разработчиками.

На рисунке 11, на примере взаимодействия двух сервисов представлен процесс самоконтекстуализации с использованием модели сервиса, представленной выше (цифрами отмечен порядок действий). Для упрощения восприятия в данном примере у сервисов выделяются только основные характеристики, имеющие непосредственное отношение к процессу самоконтекстуализации.

Сервис 1 воздействует на окружение (действие 1), внося соответствующие изменения в оперативный контекст, отражающий знания о конкретной ситуации, структура которого задана абстрактным контекстом данного сервиса (действие 2). Часть оперативного контекста сервиса 1 связана с оперативным контекстом сервиса 2 (сопоставление онтологий (действие 3) позволило определить общие классы в их оперативных контекстах). Изменение оперативного контекста сервиса 2 вызывает корректировку его поведения (действие 4) и/или структуры его абстрактного контекста (действие 5) согласно текущей ситуации и сервис начинает выполнять набор действий, заданных для исполнения в соответствующей ситуации.

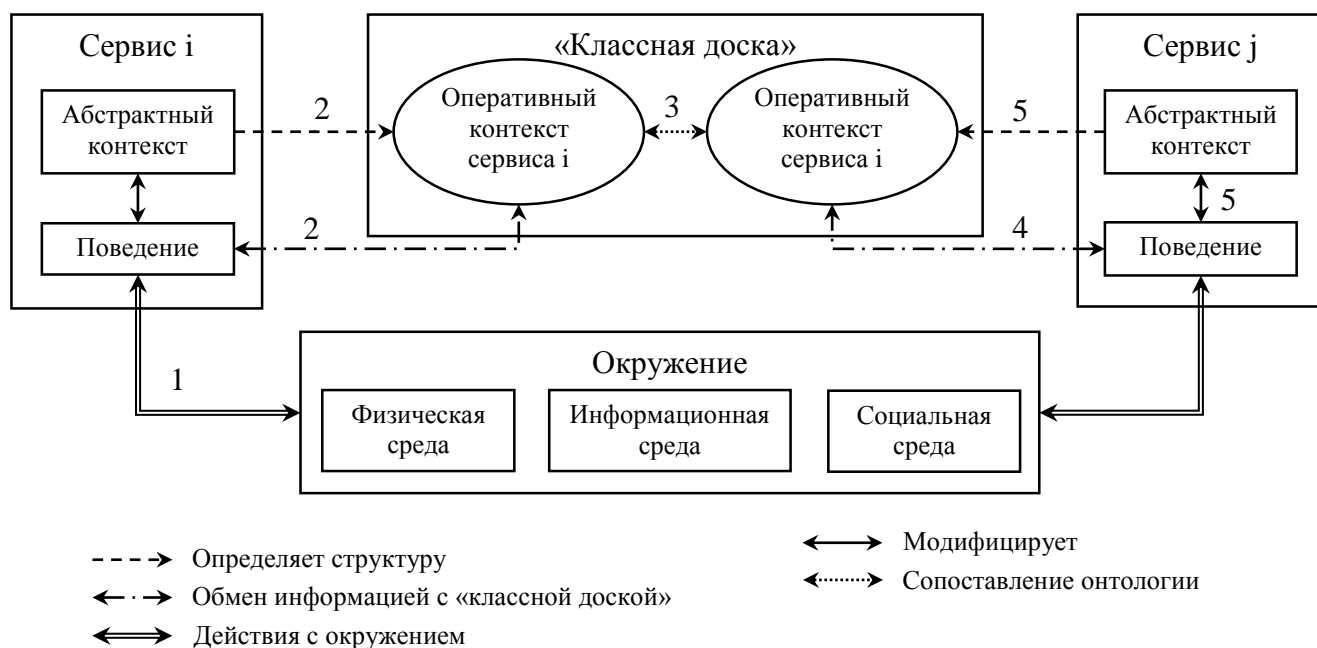


Рисунок 11 — Самоконтекстуализация сервисов в системе обеспечения инфомобильности

## 2.4 Онтологическая модель контекста для сервис-ориентированной системы обеспечения инфомобильности

Для принятия сервисами контекстно-зависимых решений и обеспечения самоконтекстуализации необходимо обеспечить формализацию контекста для его машинной обработки. Процесс моделирования ситуации для формализации контекста предлагается разделить на два этапа [21]. Первый этап основан на построении модели текущей ситуации, основанной на знаниях проблемной области. Результатом первого этапа является абстрактная модель текущей ситуации, отражающая структуру знаний проблемной области. На втором, оперативном, этапе абстрактная модель конкретизируется значениями переменных, определенных в текущей ситуации и получаемых от информационных сервисов, входящих в состав системы обеспечения инфомобильности (оперативный контекст).

В соответствии с этапами моделирования текущей ситуации предлагается использовать два типа контекста: абстрактный и оперативный. С помощью абстрактного контекста описывается онтологическая модель текущей ситуации, построенная на основании интеграции знаний проблемной области, релевантных для данной ситуации. Оперативный контекст конкретизирует абстрактный контекст согласно текущей ситуации. Подобное разделение позволяет разделить формализацию задачи и ее конкретизацию, за счет чего для каждой задачи заранее могут быть определены способы ее решения и подобраны сервисы, предоставляющие наиболее релевантную информацию.

Знания проблемных областей сервисов в абстрактном контексте формализованы с помощью онтологии, включающей в себя концептуальные знания (знания об объектах,

составляющих проблемную область), представленные посредством спецификации свойств (атрибутов) этих объектов и существующих между ними отношений; и знания о задачах, существующих в данной области, и методах их решения (онтология задач и методов). Методы реализованы как вычислительные сервисы, ссылки на которых также определены в онтологии.

Преимущества, предоставляемые онтологией, при использовании ее в качестве основной модели представления знаний в информационных системах, подробно описаны в работах [45; 59; 75; 89; 153]. Среди всех преимуществ стоит особо выделить следующие, особо актуальные для системы обеспечения инфомобильности

- Онтологии предоставляют стандартизированный способ представления знаний. Наличие стандартов существенно упрощает разработку и использование модели знаний, основанной на онтологии. Использование веб-ориентированного формата позволяет также создавать репозитории информации, которые могут быть использованы другими разработчиками.
- Онтологии позволяют повторно использовать знания. За счет предоставления дополнительной информации, такой как семантическое описание сущностей базы знаний, связи между сущностями и т.д., части онтологий могут быть применены при разработке новых онтологий в новых проблемных областях, тем самым объединяя общие понятия в разных областях.
- Онтологии упрощают обмен знаниями. При использовании онтологий в разработке семантических веб-сервисов достигается гибкость, позволяющая реализовывать динамические сценарии. Разработчики сервисов могут изменять онтологии, при этом не нарушая интероперабельности. Измененные семантические связи и описания могут быть распознаны другими сервисами за счет механизмов сопоставления онтологии, что не только упрощает обмен знаниями, но и делает систему устойчивой и способной к автоматической конфигурации своего состояния.
- Онтологии позволяют интегрировать разнородные знания и информацию. Знания и информация из разнородных независимых источников могут быть интегрированы в общую модель посредством описания их семантической связи с помощью онтологии и использоваться в дальнейшем для решения задачи, требующей учет знаний разных видов.

При формализации знаний проблемной области сервиса, описываемых прикладной онтологией, используются средства объектно-ориентированных сетей ограничений [12; 21]. В соответствии с данным формализмом онтология  $A$  представляется в следующем виде:

$$A = \langle O, Q, D, R \rangle, \quad (1)$$

где  $O$  — множество классов, представляющих объекты;  $Q$  — множество атрибутов классов, в котором выделяются три подмножества атрибутов, описывающих: параметры физической среды  $P = \{p_1, \dots, p_n\}$ ; результаты обработки информации и другие информационные составляющие  $V = \{v_1, \dots, v_m\}$ ; социальные характеристики пользователя  $S = \{s_1, \dots, s_l\}$ ,  $n, m, l \in \mathbb{N}$ , таким образом  $Q = \{P, V, S\}$ ;  $D$  — множество доменов — областей допустимых значений атрибутов;  $R$  — множество отношений (ограничений), которыми связаны классы. Отношениями описываются: таксономия классов (отношение «быть экземпляром»), иерархия классов («быть частью»), наследование, свойства классов и т.д.

При формировании абстрактного контекста текущей ситуации из множеств в онтологии  $A$  формируются соответствующие подмножества, в которые входят только те элементы исходных множеств, которые используются для описания сложившейся ситуации. При этом в онтологическую модель добавляется множество сервисов, которые могут предоставить знания и информацию для формирования контекста, и сервисов, способных выполнять определенные действия с имеющимися знаниями и информацией. Таким образом абстрактный контекст может быть выражен следующим образом:

$$Context_{abs}(C, T_{abs}) = \langle O_{abs}, Q_{abs}, D_{abs}, R_{abs}, WS_{abs}, T_{abs} \rangle \quad (2)$$

где  $C$  — моделируемая ситуация;  $O_{abs} \subseteq O$  — множество классов, требуемых в общем случае для моделирования ситуации  $S$ ;  $Q_{abs} \subseteq Q$  — множество атрибутов классов  $O_{abs}$ ;  $D_{abs} \subseteq D$  — множество доменов атрибутов  $Q_{abs}$ ;  $R_{abs} \subseteq R$  — множество ограничений, включенных в абстрактный контекст;  $WS_{abs}$  — множество Web-сервисов, моделирующих функции информационных ресурсов, присваивающих значения атрибутам  $Q_{abs}$  и способных выполнять обработку имеющейся информации,  $WS_{abs} \subseteq WS$ , где  $WS$  — множество зарегистрированных Web-сервисов;  $T_{abs}$  — прогнозируемое время адекватности модели.

Для того чтобы было возможно получение информации о текущей ситуации от сервисов, входящих в состав системы, между онтологиями и источниками устанавливаются связи, показывающие, от какого источника могут быть получены соответствующие значения свойств объекта. Так как источниками информации являются сервисы, также описываемые онтологией, то установление связи возможно с помощью механизма сопоставления онтологий, описанного в разделе 3.2.

По мере получения информации от других сервисов осуществляется присвоение значений атрибутам классов абстрактного контекста, и формирование оперативного контекста. Оперативный контекст  $Context_{op}$  является моделью текущей ситуации, описываемой



абстрактным контекстом  $Context_{abs}$ , с присвоенными значениями атрибутов  $Q_{abs}$ . Данная модель может интерпретироваться как задача удовлетворения ограничений. Модель прикладного контекста представляется как:

$$Context_{op}(C, t) = \langle O_{op}, Q_{op}, D_{op}, R_{op}, WS_{op}, T_{op}, \Delta T \rangle \quad (3)$$

где  $t$  — текущее время;  $O_{op} \subseteq O_{abs}$  — множество классов, которые реально используются для моделирования ситуации  $C$  при конкретных условиях;  $Q_{op} \subseteq Q_{abs}$  — используемые множества атрибутов;  $D_{op} \subseteq D_{abs}$  — используемые множества доменов;  $R_{op} \subseteq R_{abs}$  используемые множества ограничений;  $WS_{op} \subseteq WS_{abs}$  — используемые множества Web-сервисов;  $\Delta T = t - t_0$  — текущее время жизни оперативного контекста,  $t_0$  — время создания абстрактного контекста.

Для каждого из сервисов, входящих в состав системы обеспечения инфомобильности, разработана онтология, наиболее полно отвечающая специфике соответствующей проблемной области, примеры которых приведены в последующих подразделах. Таким образом общую онтологию системы для любой ситуации при необходимости можно сложить из частных онтологий сервисов, образованных с учетом рассматриваемой ситуации. Дополнительно приводится описание онтологии используемой на устройствах пользователя для предоставления информации сервисам системы.

### **Онтология контекста пользователя**

С помощью контекста пользователя в системе обеспечения инфомобильности описываются его основные характеристики и состояние его окружения. Данная информация используется сервисами системы для предоставления персонализированной поддержки с учетом текущей ситуации. Для отображения контекста пользователя в системе обеспечения инфомобильности была разработана онтология, представленная на рисунке 12.

Контекст пользователя состоит из двух частей: 1) информация, предоставляемая самим пользователем, его профиль (имя, предпочтения, владение транспортным средством, ключевые места на карте и т.д.) и 2) автоматически собранная информация об окружении пользователя (текущее местоположение, скорость движения, погода и т.д.).

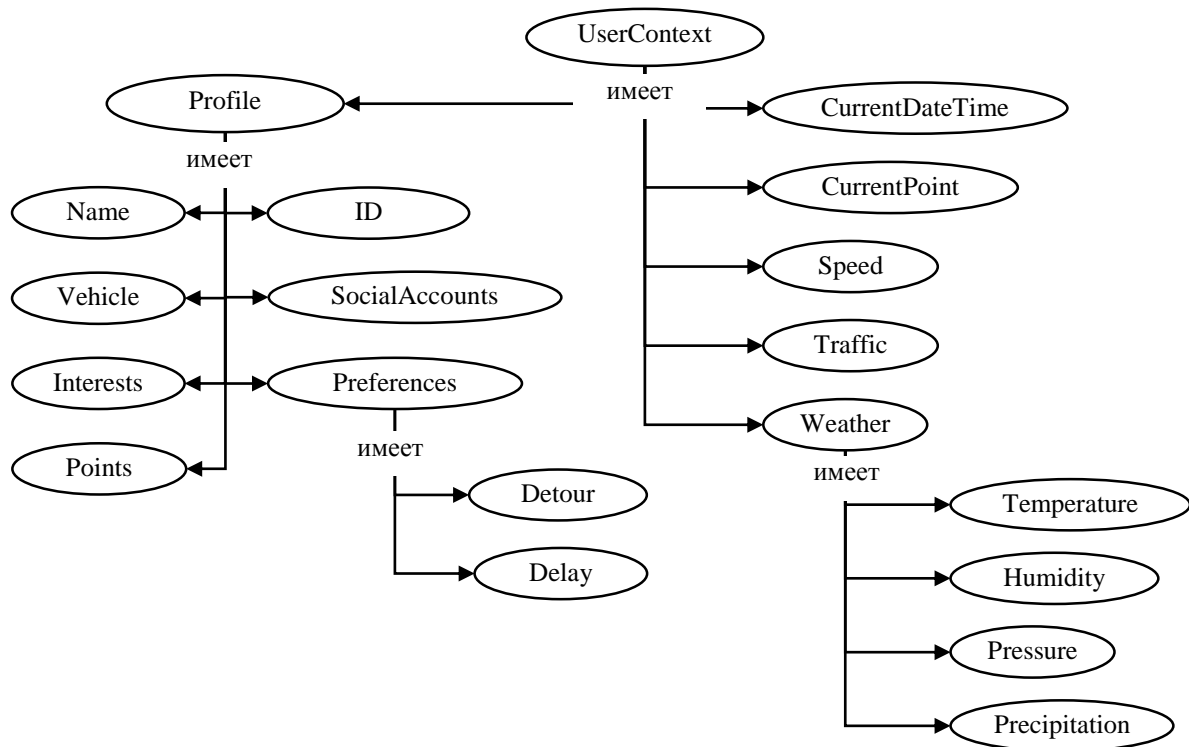


Рисунок 12 — Онтология контекста пользователя

В профиль пользователя включаются следующие атрибуты (после знака «::» определяется домен, к которому принадлежат значения атрибута):

- Name::string[256]. Имя пользователя. Вводится пользователем, либо извлекается из страницы в социальной сети, в случае, если пользователь предоставил доступ к информации из нее.
- ID::integer. Уникальный идентификатор пользователя в системе обеспечения инфомобильности. Используется для предоставления персонализированной поддержки.
- Vehicle::Vehicle. Атрибут содержит транспортное средство, закрепленное за пользователем. Онтология, описывающая ТС представлена в следующем разделе.
- Interests::string. Атрибут содержит интересы пользователя. На основе этих интересов осуществляется поиск и фильтрация объектов на карте.
- Points::Point. Закладки пользователей, связанные с географическим местоположением, например, «Дом», «Работа» и т.д.
- SocialAccounts::URL. Привязки к профилям социальной сети.
- Preferences. Предпочтения пользователя. Detour — расстояние, которое пользователь согласен пройти пешком. Delay — время, которое пользователь согласен находиться на одном месте.

К контексту пользователя относятся атрибуты, получаемые от сенсоров устройств пользователя и от сервисов-поставщиков информации:

- CurrentDateTime::timestamp, timezone. Текущее время пользователя с учетом часового пояса
- CurrentPoint::[Latitude,Longitude]. Текущие координату пользователя.
- Speed::integer. Текущая скорость пользователя.
- Traffic::integer. Состояние дорожного трафика рядом с пользователем.
- Weather::string. Погода рядом с текущим положением пользователя. Включает в себя Temperature::integer — температура, Humidity::integer — влажность, Pressure::integer — атмосферное давление, Precipitation::string — осадки.

### Онтология сервиса планирования мультимодальных поездок

Проблемная область сервиса планирования маршрутов охватывает задачи, связанные с использованием картографической, временной и персональной информации для планирования маршрутов с использованием различных видов транспорта. В связи с этим, в онтологии данного сервиса должны рассматриваться объекты, связанные с представлением картографической информации, информации о расписании транспортных средств, характеристики транспортных средств и характеристики пользователей.

Для удобства восприятия онтология сервиса планирования мультимодальных маршрутов разделена на части, раскрывающие понятия онтологии верхнего уровня, представленной на рисунке 13. В данной онтологии определены сущности, используемые для представления маршрутов в системе по всем доступным видам транспорта.

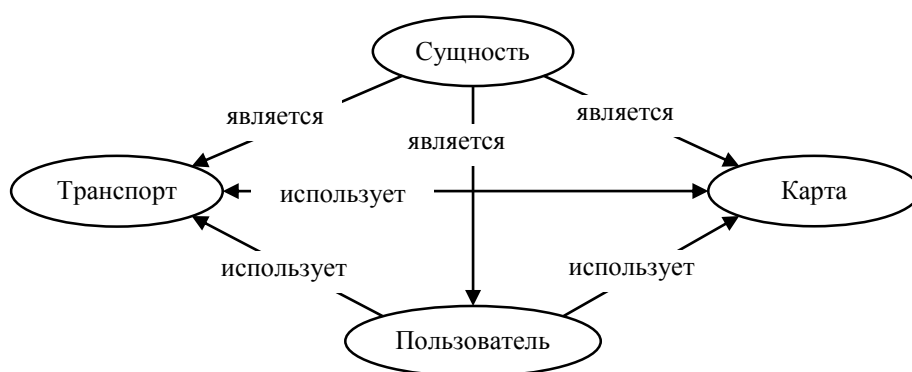


Рисунок 13 — Онтология верхнего уровня для сервиса планирования мультимодального маршрута

В онтологии верхнего уровня сервиса планирования мультимодальных маршрутов выделено три типа сущностей: транспортное средство, пользователь и карта. Пользователи системы могут использовать определенное транспортное средство и следовать на нем по

маршруту, а также осуществлять поиск маршрутов с использованием карты. При этом каждому транспортному средству соответствует определенный маршрут на карте.

Пользователь может совершать как пешую прогулку, так и пользоваться одним из видов транспортных средств, среди которых выделяются следующие: велосипед, легковой автомобиль и общественный транспорт (автобус, троллейбус, трамвай, метрополитен). Каждое из транспортных средств можно описать с помощью онтологии, представленной на рисунке 14.

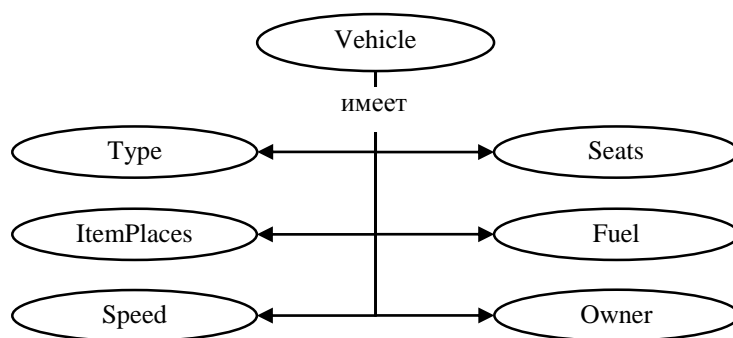


Рисунок 14 — Класс "Транспорт"

Для описания класса транспорта используются следующие атрибуты:

- **Type**. Определяет тип транспортного средства. Доменом является множество из шести элементов  $D_{Type} = \{\text{Пешеход, Автомобиль, Автобус, Троллейбус, Трамвай, Метро}\}$
- **Seats::integer[100]**. Количество мест для пассажиров
- **ItemPlaces::integer[100]**. Количество мест для багажа
- **Fuel::integer[100]**. Уровень топлива в баке автомобиля.
- **Speed::integer[120]**. Средняя скорость движения транспортного средства
- **Owner::bigint**. Если ТС – автомобиль, то для него дополнительно определяется идентификатор владельца. Для общественного транспорта поле Owner задает ID автопарка, к которому приписан конкретный транспорт
- **Route::LinkedList<Point>**. Отображает маршрут, по которому следует транспорт. Маршрут определяется упорядоченным списком точек карты.

Среди пользователей сервиса планирования мультимодальных поездок выделяются две категории: водители и пассажиры. Категория «водитель» применяется для пользователей, отметивших факт владения транспортным средством в своем профиле на мобильном устройстве. Пассажир характеризуется точками на карте, задающими его текущее положение и пункт назначения, а также дополнительными предпочтениями, такими как использование определенных видов транспорта. Класс, описывающий пользователя, представлен на рисунке 15.

Для решения задач сервиса планирования мультимодального маршрута при описании пользователей используются следующие атрибуты:

- ID::hash. Уникальный идентификатор пользователя;
- Name::string[256]. Имя пользователя;
- Type. Определяет тип пользователя из множества {Пассажир, Водитель}. Тип пользователя «Водитель» назначается только в случае построения автомобильного маршрута с использованием личного транспорта пользователя.
- Point::point. Точки, указанные пользователем. За каждым пользователем может быть закреплено несколько точек (как минимум две: начальная и конечная)
- Delay::integer. В случае построения маршрута с использованием общественного транспорта обозначает максимальное время ожидания транспортного средства на остановках. При планировании маршрута с совместным использованием личного транспорта указывает время, в течении которого пользователи согласны ждать друг друга в точке встречи.
- Vehicle. Обозначает используемое транспортное средство.
- Detour. При планировании маршрута с использованием общественного транспорта показывает максимальное расстояние между остановками, которое согласен преодолеть пользователь в случае необходимости пересадки. При планировании поездки с совместным использованием личного транспорта, отражает расстояние, которое пользователь-пассажир согласен преодолеть до точки встречи и до конечной точки после высадки, а также максимальное отклонение от кратчайшего пути для пользователей-водителей

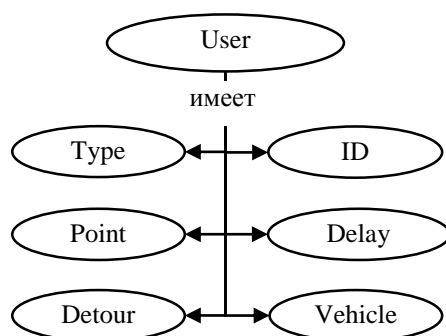


Рисунок 15 — Класс «пользователь»

Сущность «Карта» (Map) описывает все объекты, связанные с представлением карты для использования ее в сервисе планирования мультимодальных маршрутов. Данная сущность включает в себя следующие объекты «Точка» (Point) и «Маршрут» (Route). Понятия, принадлежащие сущности «Карта» и входящим в нее сущностям представлены на рисунке 16.

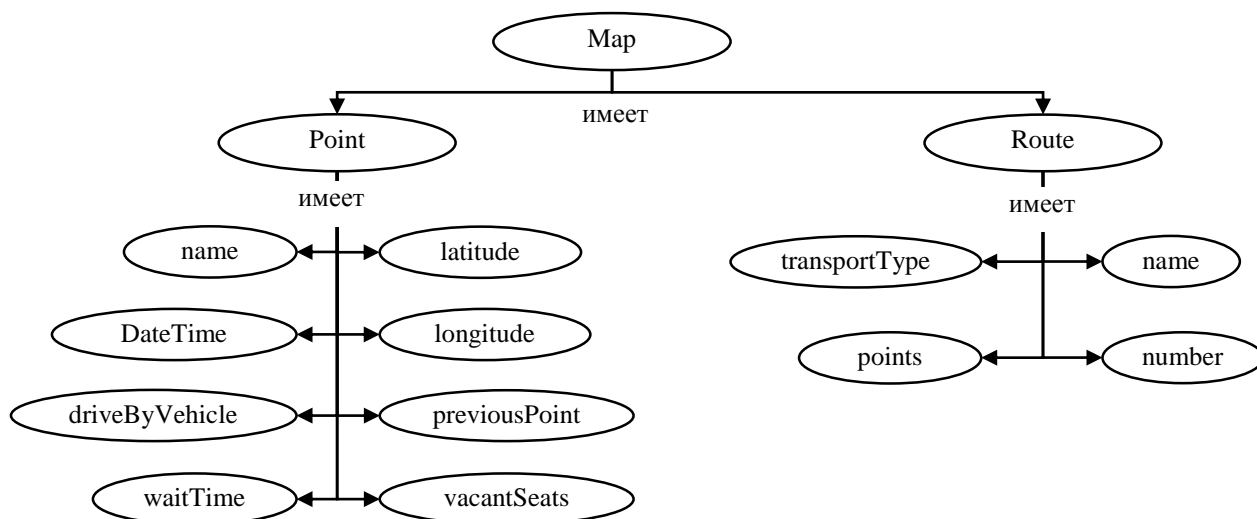


Рисунок 16 — Класс «Карта»

Класс Point («Точка») используется для описания ключевых точек в пути пользователя. В составленной онтологии класс «Точка» описывается следующими атрибутами:

- `previousPoint::bigint`. В случае, если точка является частью маршрута, содержит указатель на предыдущую точку пути. В случае, если точка является начальной, содержит значение FALSE;
- `Latitude::double`. Широта точки;
- `Longitude::double`. Долгота точки;
- `driveByVehicle::Vehicle`. Определяет конкретное транспортное средство, если точка принадлежит маршруту. В случае, если рассматриваемая точка является точкой пути пассажира, содержит указатель на водителя, который в данный момент перевозит этого пассажира.
- `vacantSeats::integer[100]`. число свободных мест в транспортном средстве в рассматриваемой точке;
- `dateTime::timestamp`. Расчетная дата прибытия пользователя в рассматриваемую точку;
- `waitTime::integer`. Время ожидания ближайшего транспортного средства в рассматриваемой точке в минутах.

В следующем разделе приводится описание онтологий для других сервисов системы обеспечения инфомобильности, которые используют фрагменты онтологии для сервиса планирования мультимодальных маршрутов и фрагменты профиля пользователя.

## Онтологии сервисов поиска объектов и информации по местоположению, сервиса выработки рекомендаций и сервиса сопоставления онтологий

Проблемная область сервисов поиска объектов и информации по местоположению, а также сервиса выработки рекомендаций охватывает задачи, связанные с использованием картографической, справочной и персональной информации для поиска и ранжирования различных объектов на карте, согласно предпочтениям пользователя и отзывам других пользователей. В связи с этим, в онтологиях этих сервисов должны рассматриваться объекты, связанные с представлением картографической информации, информации о точках на карте, предпочтениях пользователя и оценках.

Сервисы поиска объектов и информации по местоположению, а также сервис выработки рекомендаций используют общие части онтологии, связанной с географическим представлением объектов в системе обеспечения инфомобильности. Обобщенная онтология сервисов представлена на рисунке 17.

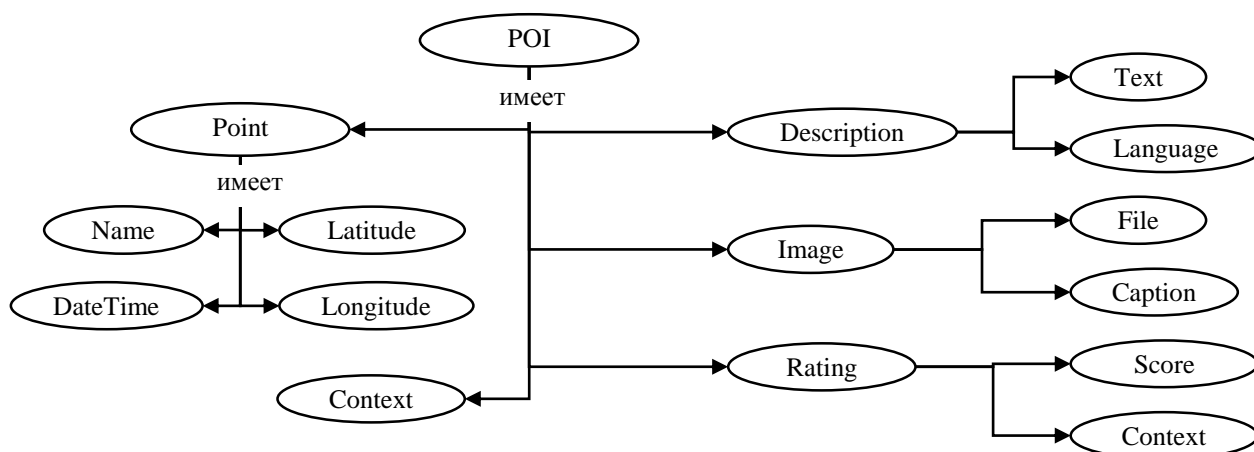


Рисунок 17 — Онтология для сервисов поиска объектов и информации

Для представления объектов (POI) используются следующие атрибуты:

- Point. Аналогичен классу point в онтологии, используемой для сервиса планирования маршрута.
- Description. Включает в себя два класса: Text::string, используемый для представления описаний объекта и Language::string, используемый для представления языка, на котором приводится описание.
- Image. Также включает в себя два класса: File::URL, содержащий ссылку на изображение и Caption::string, предоставляющий название или описание картинки.
- Rating. Используется для представления рейтинга объекта. Включает в себя: Score::float[5.0] — конкретную числовую оценку в интервале от 0 до 5 и Context,

аналогичный контексту пользователя, определяющие ситуацию, для которой актуальна полученная оценка.

- Context. Текущий контекст объекта. Структура аналогична контексту пользователя.

Для сервиса сопоставления онтологий не требуется реализация отдельной онтологии, так как он имеет возможность использовать онтологии всех сервисов системы обеспечения инфомобильности для обеспечения их сопоставления. Для этого сервисы системы размещают свой абстрактный контекст на «классной доске», а сервис сопоставления производит их интерпретацию и сопоставление с другими имеющимися онтологиями.

Сервис обеспечения безопасности конфиденциальных данных пользователя принимает решение о предоставлении доступа к данным на основе контекста других пользователей. Для сбора и обработки контекста используется соответствующая онтология контекста пользователя.

## **2.5 Выводы по главе 2**

- 1) В главе предложен подход к построению системы обеспечения инфомобильности, основанный на использовании сервисов в качестве источников информации и знаний и их обработчиков. Подход основан на следующих принципах, выведенных из требований: открытость данных и сервисов, использование онтологий, распределенность архитектуры, ориентированность на пользователя, использование контекстной информации, самоконтекстуализация сервисов, работа в режиме реального времени, мультимодальность маршрутов, конфиденциальность информации о пользователе. Следование этим принципам позволит удовлетворить основные требования и предоставить пользователю качественные услуги для обеспечения его мобильности.
- 2) Предложенный подход является сервис-ориентированным. Использование сервисов имеет ряд преимуществ среди которых выделяется автономность сервисов, их стандартизация, возможность повторного использования и композиции, что согласуется с принципами, на которых основан подход, и позволяет сделать систему масштабируемой и легко конфигурируемой.
- 3) В рамках подхода предложены сценарная, концептуальная и онтологическая модели системы обеспечения инфомобильности. Сценарная модель позволяет провести анализ основных сценариев использования системы, среди которых были выделены следующие: планирование маршрута, поиск объектов рядом с пользователем и визуализация контекстной информации. Проведение анализа



сценариев позволило определить минимально необходимый набор сервисов, обеспечивающих их выполнение.

- 4) В предложенной концептуальной модели определяются сервисы, необходимые для реализации сценарной модели и способ организации сервис-ориентированного подхода к архитектуре системы. Сервисы, определенные в концептуальной модели, являются композицией более мелких сервисов, позволяющей предоставить комплексное решение общей для них задачи. Так, в составе системы выделяются: сервис планирования мультимодальных маршрутов, сервис поиска объектов по местоположению, сервис поиска информации по местоположению, сервис обеспечения конфиденциальности информации, сервис формирования рекомендаций и сервис сопоставления онтологий. Взаимодействие перечисленных сервисов осуществляется посредством модели архитектуры «классная доска», позволяющей предоставить сервисам общее хранилище информации.
- 5) Для представления информации на классной доске используется онтология. Онтология позволяет структурировать информацию и представить ее в виде, пригодном для машинной обработки, и в то же время понятном для человека. Одним из главных преимуществ онтологии является возможность ее повторного использования как полностью, так и по частям. На этом основана самоконтекстуализация сервисов, т.е. возможность самостоятельного описания и изменения контекста. Контекст моделируется с помощью онтологии и может изменяться путем расширения онтологии, за счет включения в нее новых понятий, либо сужаться, путем исключения ненужных. Для каждого сервиса определена онтология, наиболее полно описывающая соответствующую предметную область.
- 6) Представленные модели позволяют обеспечить интероперабельность сервисов в составе системы и учет контекста при формировании решения по обеспечению информомобильности. Сервисы могут изменять модель описания знаний для более полного соответствия текущей ситуации, что позволяет адекватно реагировать на ее изменение. Методы, применяемые для построения основных сервисов системы обеспечения информомобильности, представлены в следующей главе.

## Глава 3. Методы и модели сервисов для системы обеспечения инфомобильности

### 3.1 Архитектура системы обеспечения инфомобильности

Архитектура системы обеспечения инфомобильности была разработана на основе концептуальной модели, представленной в разделе 2.3. В рамках модели было обосновано представление компонент системы в качестве независимых сервисов, взаимодействующих асинхронно посредством модели архитектуры «классная доска». Данная модель предполагает наличие общего объекта, посредством которого осуществляется обмен информацией между сервисами. При этом объект должен быть доступен для всех сервисов, независимо от их расположения, от устройств, на которых они запущены и других факторов.

В качестве технологии, которая может быть использована для организации архитектуры «классная доска» и которая удовлетворяет требованиям сервис-ориентированной системы обеспечения инфомобильности, в данной работе предлагается технология интеллектуальных пространств [9; 110]. Интеллектуальное пространство может быть определено как часть информационно-физического пространства, в котором вычислительные и информационные ресурсы, а также участники виртуальных сообществ взаимодействуют друг с другом, с целью предоставления общего доступа к имеющейся информации (рисунок 18).

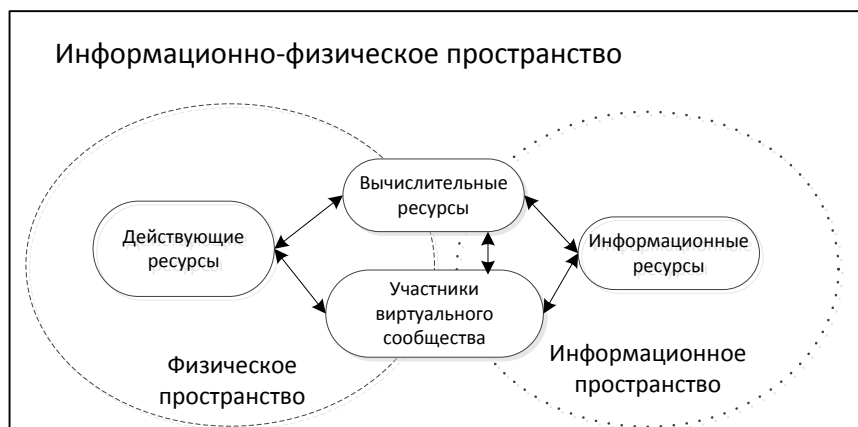


Рисунок 18 — Интеллектуальное пространство, как часть информационно-физического пространства

Идеи, лежащие в основе интеллектуальных пространств, позволяют реализовать концепцию "Интернет вещей" ("Internet of Things"). Эта концепция призвана упростить повседневную жизнь человека за счет автоматизации рутинных действий, производимой на устройствах, окружающих человека. Такими действиями, к примеру, могут быть снятия показания счетчиков, контроль освещенности в комнатах при перемещении между ними и т.п. Объединяя свои вычислительные мощности в единую сеть и распределяя между собой задачи,

устройства могут обеспечить координированную поддержку пользователя, основанную на его предпочтениях и текущей ситуации в информационном и физическом пространстве, формализованном с применением модели контекста [28]. Интеллектуальные пространства являются развитием концепции облачных вычислений и соединяют в себе идеи распределенных (повсеместных) вычислений и Семантических сетей (Semantic Web). В работе [127] представлены следующие основные идеи интеллектуальных пространств в сравнении с хорошо зарекомендовавшей себя технологией облачных вычислений (см. таблицу 3).

Таблица 3 — Сравнение основных идей облачных вычислений и интеллектуальных пространств

<b>Облачные вычисления</b>	<b>Интеллектуальные пространства</b>
Определяются поставщиком услуг	Определяется пользователем
Для пользователей предоставляется централизованно (но распределено на оборудовании провайдера)	Распределено между устройствами, составляющими интеллектуальное пространство
Требуют постоянного подключения к сети связи	Не требуют постоянного подключения к сети связи
Всегда есть проблемы, связанные с приватностью данных и их владением	Данные хранятся только на устройстве пользователя, есть некоторые проблемы, связанные с владением (распространение, цитирование, аккредитация информации)
Вычислительные ресурсы неограниченны Ресурсы для хранения данных неограниченны	Вычислительные ресурсы и ресурсы для хранения информации ограничены ресурсами устройств, составляющих интеллектуальное пространство (но могут расширяться с использованием технологии облачных вычислений)
Требуют материальных затрат	Не требуют материальных затрат
Облака не личные, контролируются поставщиком	Пространства личные, контролируются пользователем
Пользователь частично ответственен за "облако", согласно лицензионному соглашению	Пользователь несет полную ответственность за пространство
Приложения предоставляет поставщик услуг "облака"	Пользователи могут использовать любые приложения, работающие с интеллектуальным пространством
Интероперабельность только в контексте поставщика услуг	Всеобщая интероперабельность

Архитектура системы обеспечения инфомобильности изображена на рисунке 19. Согласно разработанной модели, используется сервис-ориентированный подход к построению архитектуры. Источниками и обработчиками информации и знаний являются сервисы, установленные на устройствах пользователя и на серверах, обеспечивающих работу системы обеспечения инфомобильности. Взаимодействие сервисов происходит посредством модели архитектуры «классная доска», реализованной на основе технологии интеллектуального пространства [19].

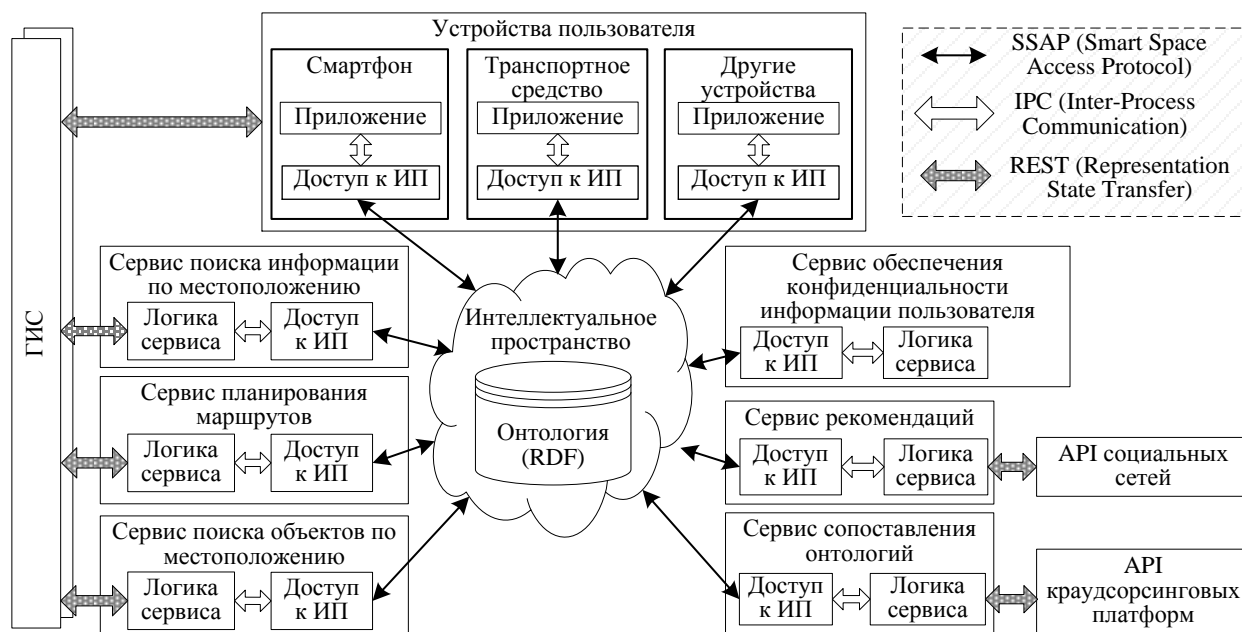


Рисунок 19 — Сервис-ориентированная архитектура СОИМ

Сервисы, размещенные на устройствах пользователя, имеют доступ к сенсорам устройств, с помощью которых определяется физическая составляющая контекста пользователя, например, текущее местоположение, локальное время и т.д. Кроме того, сервисы имеют возможность взаимодействовать с пользователем посредством визуального интерфейса, с помощью которого пользователь может передать информацию в систему и получить результаты ее обработки.

Вследствие того, что сервис планирования маршрутов, сервис поиска информации по местоположению и сервис поиска объектов по местоположению, входящие в состав системы обеспечения инфомобильности требуют обработки географической информации, для них необходима реализация механизма, позволяющего обращаться к различным геоинформационным системам, таким как погодные ГИС, ГИС для построения маршрутов и ГИС для геокодирования и обратного геокодирования. Унифицированным механизмом, позволяющим обращаться ко всем геоинформационным системам является протокол REST (Representation State Transfer). С помощью данного протокола сервисы системы обеспечения инфомобильности могут сформировать запрос и отправить его на выполнение в соответствующую ГИС, при этом с точки зрения реализации, подобное взаимодействие похоже на вызов локальной функции. Однако, ввиду большого количества запросов, требуемых для работы СОИМ, в целях снижения нагрузки на открытые ГИС была сформирована собственная ГИС с использованием открытых картографических данных. Процесс формирования ГИС и реализованные функции описаны в разделе 4.2.

Для получения информации об оценках пользователей для формирования рейтинга объектов используется обращение к социальным сетям, цель которых заключается в сборе и обработке подобной информации (TripAdvisor). Для обращения к ним также используется

протокол REST. Кроме того, оценки пользователей, выставленные с использованием интерфейса системы обеспечения инфомобильности могут сохраняться в собственной базе данных оценок сервиса, вместе с контекстом, в котором они были получены. Метод получения и обработки оценок с учетом контекста представлен в разделе 3.4.

Для обеспечения автоматизированного сопоставления онтологий с использованием технологии краудсорсинга, сервис сопоставления онтологий имеет возможность использования программных интерфейсов краудсорсинговых платформ. В данной работе используется платформа Amazon Mechanical Turk [37], на которой размещаются микрозадания на сопоставление классов онтологий. Формирование заданий и работа с выбранной краудсорсинговой платформой представлена в разделе 4.4

Для реализации технологии интеллектуального пространства используется платформа *Smart-M3* [92], имеющая открытый исходный код, описание которой представлено в следующем разделе.

### **Платформа Smart-M3**

В качестве технологической платформы для реализации концепции интеллектуального пространства используется платформа с открытым исходным кодом Smart-M3 [92; 110; 156]. Первая версия платформы была представлена на конференции NoTA, 1 октября 2010 года в Сан-Хосе. Разработка Smart-M3 проводилась по инициативе компании Nokia, в сотрудничестве с совместным предприятием ARTEMIS Joint Undertaking [41], объединяющим представителей индустриального сегмента, исследовательского сегмента и органов государственной власти из 23 стран Европейского Союза для разработки инновационных технологий в области распределенных вычислений и их применении в промышленности. Сотрудничество проводилось в рамках проекта SOFIA (smart objects for intelligent applications, умные объекты для интеллектуальных приложений) [110] и Финского национального исследовательского проекта DIEM (Device interoperability ecosystem) [103]. Целью проекта являлось создание платформы, способной соединить «информацию» из физического мира с возможностями обработки, представленными распределенными и повсеместными вычислениями [110]. Итогом исследований стала архитектура открытой информационной платформы, позволяющий обеспечить интероперабельность между различными устройствами и распределенными системами из разных доменов, реализация которой выполнена в платформе Smart-M3. На сегодняшний день платформа Smart-M3 применяется в крупных европейских проектах, требующих обеспечения взаимодействия разнородных устройств, например, проект eHealth [60], объединяющий медицинские сканеры в единую систему с возможностью предоставления краткого отчета о состоянии здоровья пользователю на мобильное устройство; eMobility [97],

предоставляющий интегрированную динамическую сетевую инфраструктуру для объединения сети передачи электроэнергии и сети управляющих устройств, позволяющую обеспечить управление распределением электроэнергии между потребителями.

Платформа Smart-M3 объединяет в себе идеи распределенных систем, повсеместных вычислений и семантических сетей. Аббревиатура M3 объединяет три ключевые идеи, лежащие в основе платформы: Multi-vendor, Multi-device, Multi-domain, то есть независимость от конкретных производителей, оборудования, области применения и возможность обмена информацией между различными программными модулями. Устройства могут быть как составлено из отдельных элементов, так и быть абсолютно независимыми друг от друга, в любом случае являясь полноценными элементами разрабатываемой системы. Обмен информацией между элементами системы осуществляется посредством интеллектуального пространства, доступного через семантический информационный брокер. Благодаря использованию семантической сети обмен информацией в интеллектуальном пространстве может осуществляться на основе протокола HTTP и с использованием унифицированных идентификаторов ресурсов (Uniform Resource Identifier — URI) [44].

Структура платформы представлена на рисунке 20. В ядре системы выделяется два основных элемента: семантический информационный брокер (*Semantic Information Broker* — SIB) и физическое хранилище данных. Посредством СИБ предоставляется простой интерфейс доступа к информационному пространству, обеспечивающий информационных агентов функциями обработки информации, такие как вставка, извлечение, редактирование, удаление и подписка на оповещения в случае изменения информации в интеллектуальном пространстве.

Информационными агентами являются программные модули, в которых реализована бизнес-логика системы. Бизнес логика включает в себя правила сбора и обработки информации, а также правила взаимодействия с интеллектуальным пространством, которое осуществляется с использованием протокола доступа к интеллектуальному пространству SSAP (*smart space access protocol*) [92]. Информация, поступающая в хранилище данных, сохраняется в виде графа, удовлетворяющего требованиям стандарта RDF (*Resource Description Framework*) [142], с помощью которого описывается семантическая сеть, в которой узлы, представляющие сущности, определенные в онтологии, и дуги, представляющие связи между сущностями, имеют унифицированные идентификаторы ресурсов. Представление онтологии системы обеспечения инфомобильности с применением стандарта RFD представлено в следующем разделе.

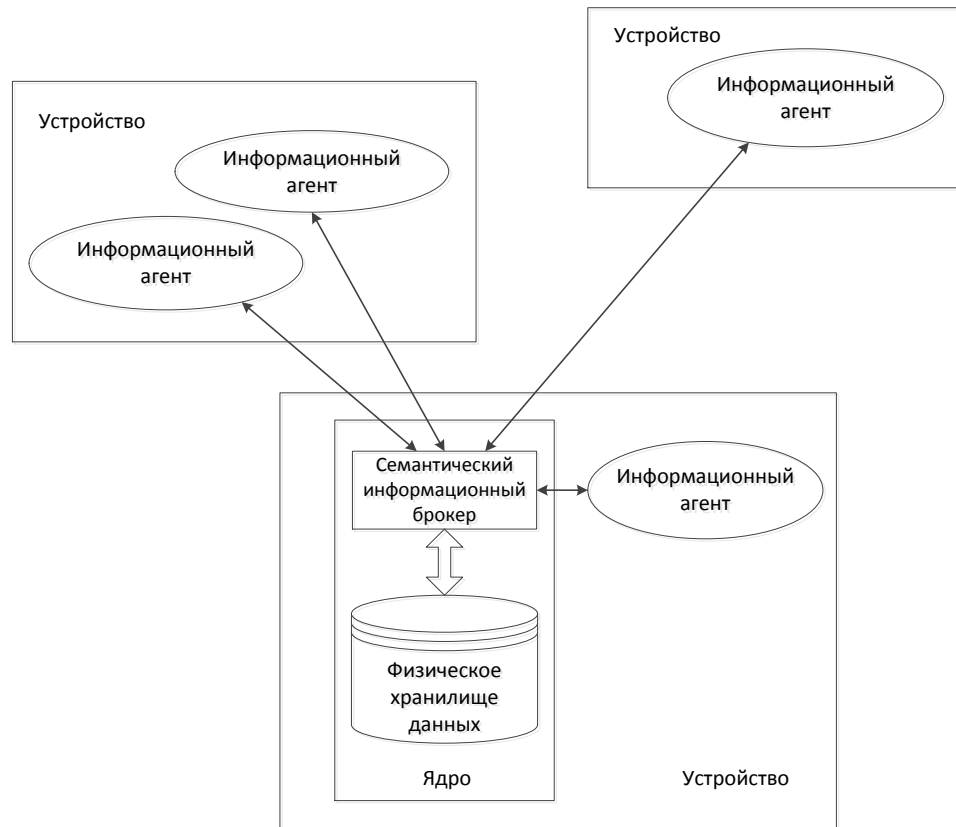


Рисунок 20 — Структура платформы Smart-M3

### Представление онтологии в интеллектуальном пространстве на базе платформы Smart-M3

Для представления информации в интеллектуальном пространстве на платформы Smart-M3 используется модель RDF [142]. Данная модель позволяет представить информацию как утверждения о ресурсах в виде, пригодном для машинной обработки и в то же время понятной для человека. Под ресурсом в RDF понимается как информационная сущность (конкретное знание или информация), так и неинформационная (человек), что позволяет использовать данную модель для представления контекста. Каждое утверждение, в соответствии с этим стандартом, описывается тремя составляющими (триплетом) «субъект – предикат – объект», позволяющими определить две сущности (субъект и объект) и тип их отношения (предикат). например, «пользователь – имеет\_время – 17-08-2015 12:36 MSK» и, по своей сути, является простым предложением. Множество подобных утверждений образует ориентированный граф, позволяющий отобразить иерархию знаний и информации. Вершинами графа являются субъекты и объекты, а дуги отображают отношения между ними.

Для представления онтологии может использоваться как язык описания онтологий OWL, так и RDFS (RDF Schema), представляющая собой специализированный словарь для определения таксономии классов. Оба языка свободно конвертируются между собой, поэтому выбор представления знаний проблемной области зависит от предпочтений разработчика.

В данной работе предлагается использовать схему RDFS, чтобы уменьшить количество необходимых трансляций между представлениями онтологии. Для взаимодействия с интеллектуальным пространством по протоколу SSAP используется представление отдельных утверждений схемы с помощью синтаксиса записи RDF, использующего язык расширенной разметки XML — RDF/XML.

Для извлечения информации из базы знаний, использующей модель RDF, консорциумом W3C разработан специализированный язык запросов и соответствующий протокол SPARQL [162; 163]. Язык запросов позволяет сформировать поисковый запрос, по синтаксису похожий на язык SQL. Протокол SPARQL подчеркивает веб-ориентированность используемой модели и открытый характер получаемых моделей знаний, требуя обязательное предоставление точек доступа, поддерживающих протокол SPARQL, которые позволяют пользователям осуществлять запросы к базе знаний. Результат запроса представляет собой набор триплетов, соответствующих тексту запроса, например, для извлечения контекста пользователя результат будет иметь следующий вид:

**Пользователь:**

(‘user1’,	‘#profile’	‘user1profile’)	‘user1’	—	условный идентификатор пользователя
‘user1profile’,	‘#name’,	‘Arthur Dent’)			
(‘user1profile’,	‘#is_a’,	‘Driver’)			
(‘user1profile’,	‘#vehicle’,	‘user1vehicle’)			
(‘user1profile’,	‘#preferences’,	‘user1preferences’)			
(‘user1preferences’,	‘#detour’,	‘300’)			Метров
(‘user1preferences’,	‘#delay’,	‘10’)			Минут
(‘user1profile’,	‘#interest’,	‘museums’)			
(‘user1’,	‘#timestamp’,	’28.12.2015 10:33:01’)			
(‘user1’,	‘#speed’,	’56’)			Км/ч
(‘user1’,	‘#traffic’,	’3’)			Балла
(‘user1’,	‘#weather’,	‘user1weather’)			
(‘user1weather’,	‘#temperature’,	‘14’)			°С
(‘user1weather’,	‘#humidity’,	‘75’)			%
(‘user1weather’,	‘#pressure’,	‘761’)			мм ртутного столба
(‘user1weather’,	‘#precipitation’,	‘2’)			облачно
(‘user1’,	‘#has_point’,	‘user1point1’)			- координаты дома



**Описание точек пользователя:**

```

('user1',      '#has_point',      'user1home')      - координаты дома
('user1home',  '#previousPoint',  'FALSE')
('user1home',  '#latitude',       '59.922669000')
('user1home',  '#longitude',     '30.252097000')
('user1home',  '#date',          '28.12.2015 10:33:01')
('user1home',  '#vacantseats',   '3')
('user1home',  '#vacantitemplaces', '2')
('user1',      '#has_point',      'user1work')      – координаты работы
('user1work',  '#previousPoint',  'user1home')
('user1work',  '#latitude',       '59.954150000')
('user1work',  '#longitude',     '30.482770000')
('user1work',  '#dateTime',     '28.12.2015 12:33:01')
('user1work',  '#vacantseats',   '3')
('user1work',  '#vacantitemplaces', '2')

```

Для упрощения восприятия формат RDF/XML приведен без дополнительных тегов, задающих разметку. Как нетрудно заметить, представленный результат запроса легко преобразуется в иерархическую структуру, соответствующую онтологии контекста пользователя, которая затем может использоваться для решения задач сервисов [151].

### **3.2 Автоматизированный метод сопоставления онтологий с использованием технологии краудсорсинга**

Для того, чтобы проанализировать существующие методы сопоставления онтологий был проведен соответствующий обзор, охватывающий системы, подходы и проекты, связанные с сопоставлением онтологий [26]. Среди них следует отметить следующие: GLUE System [39][69], Falcon-AO [94], MLMA [36], Novy [93], SKAT [117], ONION [116], Promt [125], H-Match [58], CTX-MATCH [154], SMART [126], Cupid [114], COMA [42], Similarity Flooding Algorithm [115], AgreementMaker [62], Pattern Based Approach [146], MinSMATCH [84], OntoView [102], Chimaera [114], VITRUVIUS [51][52], SAMBO [106], Falcon [95], DSSim [123], RiMOM [109], ASMOV [98], Anchor-Flood [88]. Наиболее интересные техники из представленного списка описаны далее.

Платформа VITRUVIUS объединяет множество сенсоров и поддерживает различные конфигурации сенсоров, позволяя приложениям устанавливаться динамически и работать одновременно. Преимуществом платформы является предоставление возможности повторного использования и развития существующей сети сенсоров и приложений, а также их расширения

путем добавления новых сенсоров и приложений. Для достижения семантической интероперабельности между различными компонентами используется подход. Основанный на применении онтологий. Авторы понимают онтологию как средство, унифицирующее данные, предоставленные различными компонентами системы в универсальное представление. [52]. Сопоставление онтологий используется для перевода или интерпретации из локальной онтологии (локального синтаксиса и структурное представление) в онтологию приложения и наоборот. С точки зрения разработки, использование сопоставления онтологий позволяет упростить добавление нового компонента к платформе (например, драйвера сенсора). Это не требует изменения реализации компонент, необходимо только обновить спецификацию отношения локальной терминологии и структуры, отвечающей новому сенсору которая может быть представлена в терминах онтологии приложения. Отображение между онтологией приложения и локальной онтологией осуществляется компонентом отображения, который также предоставляет интерфейсы для обмена данными и контроля (например, конфигурация сенсоров). Компонент отображения реализован как сервис для ОС Android, который может быть быстро реализован с использованием общих шаблонов разработки.

SAMBO является системой для сопоставления и слияния биомедицинских онтологий. Он работает с онтологиями, представленными в формате OWL, предоставляя точное соответствие между понятиями онтологий и отношениями между ними. Система использует различные механизмы сопоставления, включая следующие:

Терминологические: модель n-грамм, редакторское расстояние, сравнение списка слов из которых составлены термины понятий онтологий. Результаты работы этих механизмов комбинируются суммированием с использованием предопределенных весов для каждого из результатов.

Структурные, посредством итеративного алгоритма проверяющие, находятся ли проверяемые понятия на одинаковых позициях, согласно иерархиям «is-a» и «part-of» относительно уже сопоставленным понятиям, руководствуясь предположением, что проверяемые понятия также скорее всего будут совпадать.

Основанные на знаниях, использующие отношение между совпавшими сущностями в системе унифицированного медицинского языка Unified Medical Language System (UMLS) и корпус знаний, собранных из опубликованной литературы с использованием наивного байесовского классификатора.

Результаты, полученные в ходе работы этих моделей комбинируются с учетом весов, определенных пользователем. Затем осуществляется фильтрация, основанная на применении пороговых значений, результаты которой отображаются для пользователя с целью получения подтверждения. Проведя процедуру сопоставления, система может объединить совпавшие

онтологии, рассчитать результат, проверить заново созданную онтологию на связность и произвести другие операции для проверки.

Falcon представляет собой автоматический подход к сопоставлению онтологий, основанный на разделении онтологии на части. Он работает с онтологиями в форматах RDFS и OWL. Подход был специально разработан для работы с большими онтологиями и включает в себя три фазы:

- 1) Разделение онтологий
- 2) Сопоставление блоков
- 3) Поиск совпадений

Первая фаза начинается с разделения каждой онтологии на отдельные сущности, основанного на структуре (выделение классов и свойств) с созданием множеств маленьких кластеров. Разделение основано на структурной близости между классами и свойствами, например, насколько близко расположены классы в иерархии отношений `rdfs:subClassOf` и на расширении алгоритма кластеризации агломераций Rock [87]. Затем из этих кластеров собираются блоки. На второй фазе блоки онтологий сравниваются с привязками (парами сущностей, для которых соответствие было найдено заранее), например, чем больше привязок найдено между двумя блоками, тем более похожими они являются. В свою очередь, привязки обнаруживаются путем сравнения сущностей с помощью техники сравнения строк I-SUB [165].

DSSim является средой для сопоставления онтологий с использованием агентов. Система обрабатывает масштабные онтологии в форматах OWL и SKOS, и вычисляет совпадение по модели 1:1 с отношениями равенства и категоризации между понятиями и свойствами. Он использует теорию Демпстера-Шафера в контексте ответов на запросы. Конкретно, каждый агент составляет убеждение за корректность определенного соответствия гипотезы. Затем эти убеждения комбинируются в единое более последовательное представление для улучшения качества соответствия. Онтологии первоначально разделяются на фрагменты. Каждое понятие или свойство первого фрагмента онтологии рассматривается как запрос, которых расширяется на основе гиперонимов из базы знаний WordNet, рассматриваемой в качестве фоновых знаний. Эти гиперонимы используются как переменные в гипотезе для повышения убежденности. Расширенные понятия и свойства сравниваются синтаксически с аналогичными понятиями и свойствами второй онтологии для того, чтобы идентифицировать соответствующий фрагмент графа второй онтологии. Затем граф запроса первой онтологии сравнивается с соответствующим фрагментом графа второй онтологии. Для этой цели используются различные меры терминологического сходства, такие как расстояния Монгера-Элкана и расстояние Джаккарда, объединенные по правилу Демпстера. Сходства рассматриваются как различные эксперты в

теории доказательств и используются для оценки количественных значений сходства (преобразованное в функцию веса убеждения), которые заполняют матрицы сходства. Полученные соответствия выбираются на основе функции убеждения более высокого уровня для смешанных доказательств. Возможные конфликты между убеждениями решаются с помощью нечеткого подхода к голосованию, оснащенного четырьмя специальными правилами «если-то». Система не имеет отдельного пользовательского интерфейса, но использует таковой из системы ответов на вопросы AQUA, способную обрабатывать запросы на естественном языке.

RiMOM является динамической средой для сопоставления онтологий с использованием нескольких стратегий. Он сосредоточен на комбинировании нескольких стратегий сопоставления, через минимизацию риска Байесовского принятия решения и количественно предполагает характеристику сходства для каждого задания сопоставления. Эти характеристики используются для динамического выбора и комбинирования нескольких методов сравнения. В среде используются следующие базовые методы:

- 1) Лингвистическое сходство (редакторское расстояние термов сущностей, векторное расстояние между комментариями и примерами сущностей)
- 2) Структурное сходство (вариация Лавинного сходства [115] реализованная как три стратегии распространения сходства: концепт-концепт, свойство-свойство и концепт-свойство)

В свою очередь, стратегия выбора использует факторы сходства ярлыков и структуры, полученные в результате предобработки онтологий, которые требуется сопоставить, для того, чтобы определить какая информация должна быть задействована в процессе сопоставления. А именно, стратегия выбора динамически регулирует выбор конкретного свойства для лингвистического сравнения, комбинацию весов для комбинирования сравнений и выбор конкретной стратегии распространения сходства.

Automatic Semantic Matching of Ontologies with Verification (ASMOV) описывает автоматический подход к сопоставлению онтологий, который используется для интеграции информации в области биоинформатики. В целом, в подходе можно выделить два основных шага:

- 1) Вычисление сходства
- 2) Семантическая проверка

Для сопоставления алгоритму предоставляются две онтологии в формате OWL. Результатом сопоставления является модель сопоставления между сущностями онтологий. На первом шаге используется лексический анализ (сходство строк, вычисление расстояния

Левенштейна), структурный анализ (взвешенная сумма сходства домена и расстояния) и экстенциональный механизм, итеративно вычисляющий меру сходства между онтологиями, объединенными в одну общую онтологию как взвешенное среднее. Он также использует несколько общих и зависимых от предметной области источников знаний, таких как WordNet и UMLS для предоставления большей ясности при вычислении сходства. Затем он выводит сходство и проверяет его на несовместимость. Совместимость проверяется на основе шаблонов, например, вместо использования полного решения, система распознает наборы соответствий, которые гарантированно приводят к несоответствию. Процесс семантической верификации рассматривает пять типов шаблонов, например, разделенно-категоризированная противоположность, категоризация неполноты и т.д. Этот процесс совпадения повторяется с полученной моделью сходства каждый раз до тех пор, пока не будет найдено новых соответствий.

AgreementMaker – это система, включающая в себя широкий спектр автоматических способов сопоставления онтологий, модульную архитектуру, многоцелевой интерфейс, набор вычислительных стратегий и различные ручные способы, например, визуальное сравнение и полуавтоматические функции, такие как обратные отзывы. Она была разработана для поддержки больших и сложных онтологий, основанных на требованиях, исходящих из разных предметных областей, таких как пространственная и биомедицинская область. Система обрабатывает онтологии в форматах XML, RDFS, OWL, N3 предоставляя результат в виде нескольких моделей сопоставления: 1:1, 1:m, n:1, n:m. В общем, процесс сопоставления организован в двух модулях: вычисление сходства и выбор совпадений. Система комбинирует методы сопоставления на трех уровнях:

Методы, применяемые на первом уровне, сравнивают свойства концептов, такие как названия, комментарии, примеры, которые представлены как вектора формата TF\_IDF, используя косинусную меру сходства и другие меры сравнения строк, такие как редакторское расстояние или подстроки.

Второй уровень использует структурные свойства онтологии и включает две модели сопоставления, называемые наследование сходства потомками и (если два понятия совпадают с высокой степенью сходства, то сходство между потомками этих понятий также возрастает) и вклад в сходство между узлами одного уровня (которое использует отношение между понятиями одного уровня)

На третьем уровне вычисляется линейно взвешенная комбинация результатов работы предыдущих двух уровней. Результаты работы третьего уровня проходят через фильтры пороговых значений и желаемых мощности выходных множеств соответствия. Система обладает сложным пользовательских интерфейсом, глубоко интегрированным с вычислением качества

сопоставления онтологий, будучи встроенной частью процесса поиска совпадений, что позволяет пользователям получить больший контроль над ним.

Для сопоставления онтологий сервисов, представленных в формате RDF, в данной работе предлагается использовать модификацию метода многоуровневого сопоставления, предложенного в работе [158]. Данный метод был выбран в связи со спецификой реализации архитектуры системы обеспечения инфомобильности, приведенной в главе 3. Необходимость модификации метода связана с тем, что в ходе обработки результатов его работы были выявлена неопределенность, связанная с определением пороговых значений коэффициентов сходства, при неправильном выборе которых, заведомо совпадающие понятия онтологий могли быть помечены как несовпадающие и наоборот. Анализ метода, показывающий данную неопределенность представлен далее.

Метод многоуровневого сопоставления онтологий позволяет «на лету» определить соответствие понятий различных онтологий за счет их сопоставления на двух уровнях: лингвистическом и контекстном (рисунок 21).

Первым этапом в модели сопоставления является лингвистический, связанный с поиском совпадающих понятий, проводимом на основе их синтаксического и семантического сходства. Подобный поиск становится возможным благодаря тому, что онтология имеет словарь терминов проблемной области, выраженных с использованием естественного языка. Для определения синтаксического сходства используется метод «нечеткого» сравнения строк, результатом которого является нормализованное значение коэффициента синтаксического сходства  $w_{synt} \in [0,1]$  [158].

Семантическое сходство определяется с использованием открытых словарей (Wiktionary или WordNet), предоставляющих список синонимов, ассоциированных слов, определений и форм слова. В методе определены следующие коэффициенты семантического сходства [158]:

$$w(t_i, t_j) = \begin{cases} 0,5 & \text{— если } t_i, t_j \text{ синонимы} \\ 0,3 & \text{— если } t_i \text{ ассоциируется с } t_j \\ \infty & \text{— если } t_i \text{ и } t_j \text{ совпадают (одно и то же слово)} \end{cases} \quad (4)$$

где  $t_i, t_j$  — термы онтологии  $i$  и онтологии  $j$  соответственно,  $w(t_i, t_j)$  — коэффициент семантического сходства термов.

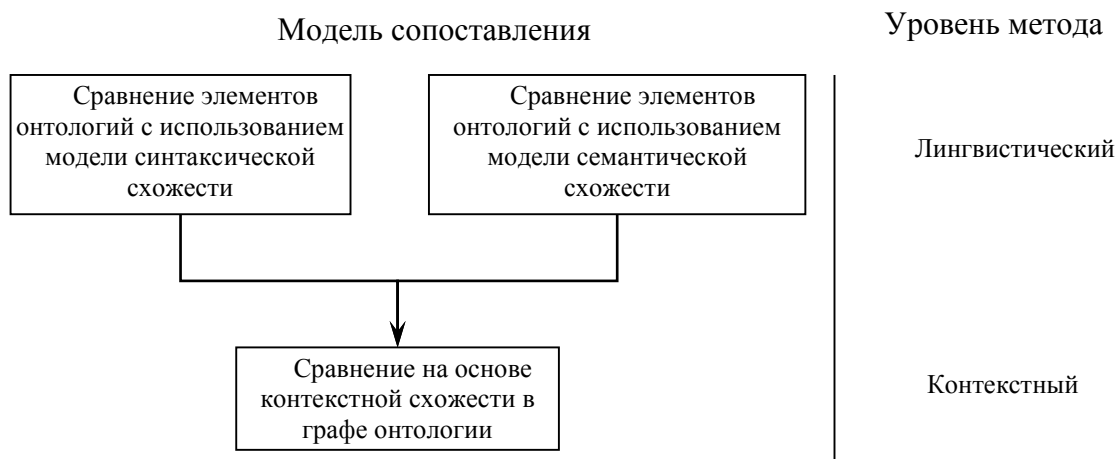


Рисунок 21 — Метод сопоставления онтологий

Вычисленные коэффициенты используются для вычисления семантического расстояния между терминами, задающими понятия онтологий. Для этого в методе применяется следующая формула:

$$dist(t_i, t_j) = \frac{1}{\sum_S \prod_{k=s_i}^{s_j} w_k}, \quad (5)$$

где  $t_i, t_j$  — термины онтологии  $i$  и онтологии  $j$  соответственно,  $w$  — коэффициент (вес) лексического соотношения, существующего между  $t_i$  и  $t_j$ ,  $S$  — множество связей, образованных между  $t_i$  и  $t_j$ , проходящих через любое количество промежуточных термов. Таким образом, за счет наличия обратной пропорции и с учетом выбранных коэффициентов для семантического сходства, семантическое расстояние будет тем меньше, чем ближе слова расположены друг к другу.

На контекстном уровне рассматриваются вычисленные семантические расстояния для определения похожих понятий онтологии с использованием графа, представляющего модель онтологии. Благодаря тому, что модель описана в формате RDF [142] (см. 2.4 и 0) становится возможным вычисление на основе семантических расстояний не только термов сопоставляемых онтологий, но и предикатов, используемых для задания отношений между понятиями, представленными этими терминами.

Вычисления на основе моделей онтологий позволяют учесть контекст понятий, улучшая качество сопоставления. Контекст учитывается за счет рассмотрения отдельных субъектов  $o_i, o_j$  в их совокупной связи с объектами  $o'_i, o'_j$  и отношениями  $r_i, r_j$  между ними (см. рисунок 22). При этом возможны два варианта пересчетов: 1) если семантическое расстояние  $dist(r_i, r_j)$  между предикатами  $r_i, r_j$  больше необходимого порогового значения  $Tr$ , то осуществляется пересчет семантического расстояния для объектов  $o'_i, o'_j$ :

$$\text{dist}(o'_i, o'_j) = \sqrt{\text{dist}(o_i, o_j) \cdot \text{dist}(o'_i, o'_j)} \quad (6)$$

2) если семантическое расстояние  $\text{dist}(o'_i, o'_j)$  между объектами онтологий больше заданного порогового значения  $Tr$ , то осуществляется пересчет семантического расстояния для предикатов:

$$\text{dist}(r_i, r_j) = \sqrt[3]{\text{dist}(o_i, o_j) \cdot \text{dist}(o'_i, o'_j) \cdot \text{dist}(r_i, r_j)} \quad (7)$$

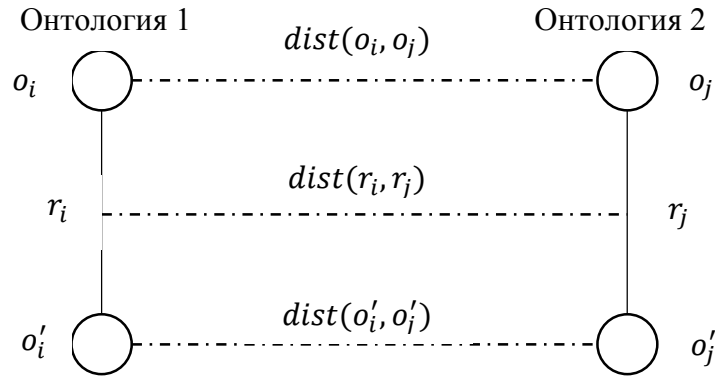


Рисунок 22 — Сопоставление моделей двух онтологий

Итоговый результат сопоставления имеет вид матрицы размером  $m \times n$ ;

$$\text{Distances}_{mn} = \begin{pmatrix} \text{dist}(o_1, o_1) & \cdots & \text{dist}(o_n, o_1) \\ \vdots & \text{dist}(o_i, o_j) & \vdots \\ \text{dist}(o_1, o_m) & \cdots & \text{dist}(o_n, o_m) \end{pmatrix}, \quad (8)$$

где  $m, n \in \mathbb{N}$ , — количество понятий сопоставляемых онтологий (размеры словарей онтологий), а элементы матрицы — вычисленные значения семантического расстояния между понятиями онтологий. Для определения похожих понятий элементы матрицы сравниваются с пороговым коэффициентом, задающим минимальное значение семантического расстояния, при котором понятия считаются совпавшими. Пороговый коэффициент рекомендуется определять после вычисления семантического расстояния классов, для исключения ситуаций, связанных с невозможностью сопоставления онтологий из-за завышенного порога семантического расстояния, либо ситуаций, связанных с нахождением совпадения между всеми классами из-за заниженного порога.

Следует отметить, что задание порога семантического расстояния требует глубокого понимания принципов работы метода и проблемных областей сопоставляемых онтологий, в связи с чем данная задача должна решаться экспертами в соответствующих областях знаний. При этом, даже не смотря на присутствие экспертов, задание порога не гарантирует как сопоставление всех понятий, которые должны быть сопоставлены, так и отсутствие понятий для которых



сопоставление не должно было пройти. В связи с этим, для корректной обработки ситуаций, связанных с неточным пороговым значением далее предложен метод автоматизированного сопоставления онтологий, согласно которому классы, для которых не удалось провести автоматическое сопоставление с использованием вышеописанного метода, предлагается отправлять на ручное сопоставление с применением технологии краудсорсинга. Краудсорсинг — это практика получения необходимых сервисов идей или контента путем распределения задачи между большой группой людей, входящих в интернет-сообщество [63].

Применительно к сопоставлению онтологий использование технологии краудсорсинга заключается в формировании микрозаданий, включающих в себя понятия из разных онтологий, с предложением оценить их семантическую близость и распределение подобных микрозаданий между членами интернет-сообщества. Из описания метода сопоставления онтологий был выведен следующий критерий, согласно которому автоматическое сопоставление считается невозможным: класс  $o_i$  онтологии 1 невозможно сопоставить с классами онтологии 2, если для всех элементов столбца  $i$  матрицы *Distances* выполняется условие:

$$\forall j (dist(o_i, o_j) < Tr), \quad (9)$$

где  $i$  – индекс класса онтологии 1,  $j$  индекс классов онтологии 2,  $dist(o_i, o_j)$  – семантическое расстояние между понятиями онтологий, полученное в результате работы метода автоматического сопоставления онтологий,  $Tr$  – пороговое значение коэффициента сходства, при котором понятия считаются похожими. Смысл этого условия заключается в том, что на автоматизированное сопоставление передаются только те понятия онтологии, для которых не нашлось ни одного понятия в другой онтологии с семантическим расстоянием больше необходимого порога. Процесс формирования заданий представлен на рисунке 23.

Условие (9) выполняется для понятия  $o_3$  онтологии 1. После проверки данного факта в автоматическом режиме формируются задачи, состоящие из всех возможных комбинаций понятия  $o_3$  онтологии 1 и понятий онтологии 2, вычисленного семантического расстояния и порогового значения. Задачи размещаются на сайте, реализующем технологию краудсорсинга и зарегистрированным участникам предлагается выставить оценку семантического расстояния  $dist(o_3, o_j)$ . Исполнители из числа зарегистрированных членов сайта независимо друг от друга выставляют семантические расстояния, которые усредняются по формуле (10), для получения итогового значения семантического расстояния.

$$dist(o_3, o_j) = \frac{\sum_{l=1}^m dist_l(o_3, o_j)}{m} \quad (10)$$

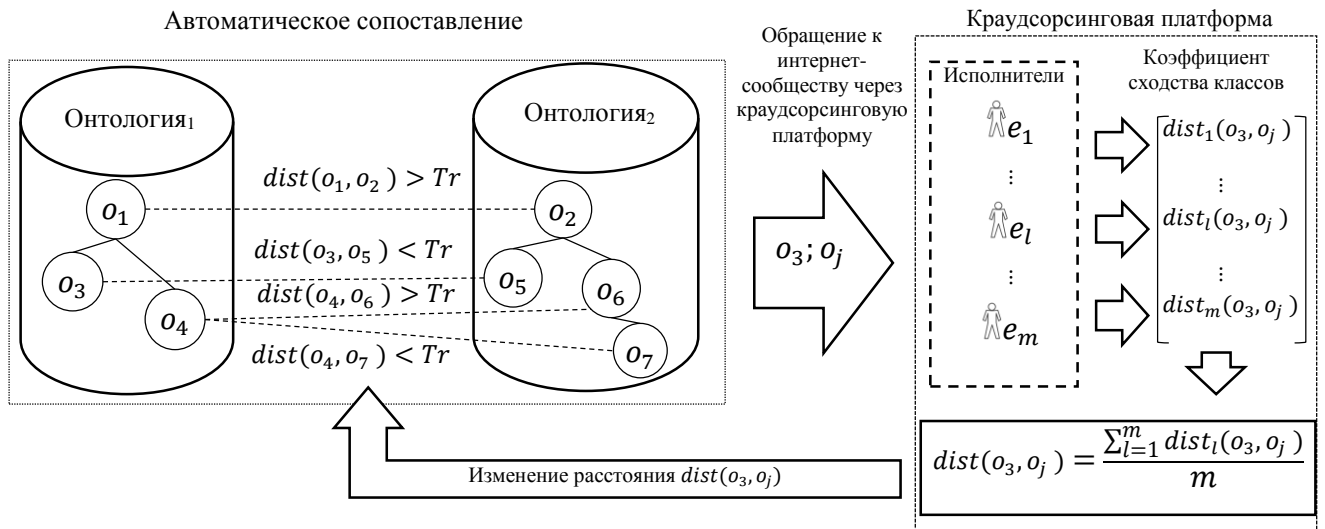


Рисунок 23 — Метод краудсорсинга для автоматизированного сопоставления онтологий

Новые значения семантического расстояния также сравниваются с пороговым значением и принимается окончательное решение о том, сопоставимы ли рассматриваемые понятия онтологий. До получения значений семантического расстояния от интернет-сообщества функциональность сервиса исключается из числа доступных в сложившемся контексте.

За сопоставление онтологий в СОИМ отвечает соответствующий сервис, реализующий описанный выше метод. Сервисы системы могут разместить свою онтологию на «классной доске» и получить модели сопоставления с онтологиями других сервисов. Полученные модели сохраняются в сервисе, запрашивающем сопоставление, и используются для получения знаний с «классной доски», согласно онтологии сервиса.

### 3.3 Метод планирования мультимодальных маршрутов

#### Планирование маршрута с использованием общественного транспорта

Как было упомянуто в описании концептуальной модели сервис-ориентированной системы обеспечения инфомобильности, сервис планирования мультимодальных маршрутов является мета-сервисом, объединяющим в себе несколько мелких сервисов, позволяющих планировать маршрут по одному виду транспорта: пешеходный, велосипедных, автомобильный и общественный транспорт. Их совместное использование позволяет предоставить услуги поиска маршрута как по одной модальности (одному виду транспорта), так и по нескольким модальностям, обеспечивая смену видов транспорта в ходе следования по маршруту. При этом транспорт может изменяться как внутри одной модальности (например, смена маршрутов общественного транспорта), так и между модальностями (например, пересадки между дальними остановками с использованием велосипеда или аренды автомобиля).

Планирование маршрутов с использованием одной модальности может быть реализовано с использованием существующих сервисов, использующих алгоритмы поиска кратчайшего пути, согласно заданным критериям. Примеры и описание особенностей реализации подобных сервисов для планирования пешеходного, велосипедного и автомобильного маршрута представлены в разделе 4.1.

В основе планирования мультимодального маршрута лежит использование общественного транспорта как основного типа транспортного средства, применяемого для совершения поездки. Все остальные типы транспорта используются для перемещения между остановками, не связанными маршрутом общественного транспорта, либо если применение другого типа транспорта предоставит выигрыш по заданному критерию оптимальности маршрута. Подобный подход позволяет перераспределить транспортный поток таким образом, чтобы больший интерес со стороны жителей города уделялся общественному транспорту, что влечет за собой снижение количества транспорта на улицах города, а вместе с тем плотность и количество заторов.

При планировании маршрута с использованием общественного транспорта необходимо учитывать расписание движения транспортных средств и текущую ситуацию на дорогах. Учет расписания позволяет оценить время, требуемое на ожидание транспортного средства в случае фиксированного и строго соблюдаемого расписания и произвести начальное планирование пути. Учет текущей ситуации уточняет первоначальный план, позволяя оценить время, требуемое на пересадку с учетом времени ожидания ближайшего транспортного средства на остановке ОТ.

В сервисе планирования мультимодальных маршрутов учет расписания движения ОТ осуществляется как для локальных маршрутов внутри города, так и для междугородних. Для планирования маршрутов между населенными пунктами используется сервис Яндекс.Расписание. Данный сервис позволяет осуществлять поиск прямых маршрутов с учетом даты и времени поездки, дополнительно оповещая о стоимости поездки. Сервис предоставляет возможность формирования запросов по протоколу REST, позволяя автоматизировать формирование запросов и обработку ответов.

Данные о маршрутах и расписаниях общественного транспорта в городах предоставляются соответствующими органами местной власти в формате GTFS (General Transit Feed Specification) [80]. В настоящее время данный формат не является единственным для представления данных о маршрутах, однако удобство его применения и расширяемость позволяют предположить, что в ближайшее время он заменит большинство сторонних форматов.

Сеть маршрутов общественного транспорта, полученную в формате GTFS можно представить в виде ориентированного взвешенного мультиграфа  $G(V, E, L)$ , в котором вершины  $V = \{v_1, \dots, v_n\}$ ,  $n \in \mathbb{N}$  соответствуют остановкам ОТ, а ребра  $E = \{e_1, \dots, e_m\}$ ,  $m \in \mathbb{N}$  —

маршрутам, соединяющим эти остановки,  $L$  – веса ребер  $E$ . Количество ребер, соединяющих вершины, соответствует количеству маршрутов между соответствующими остановками.

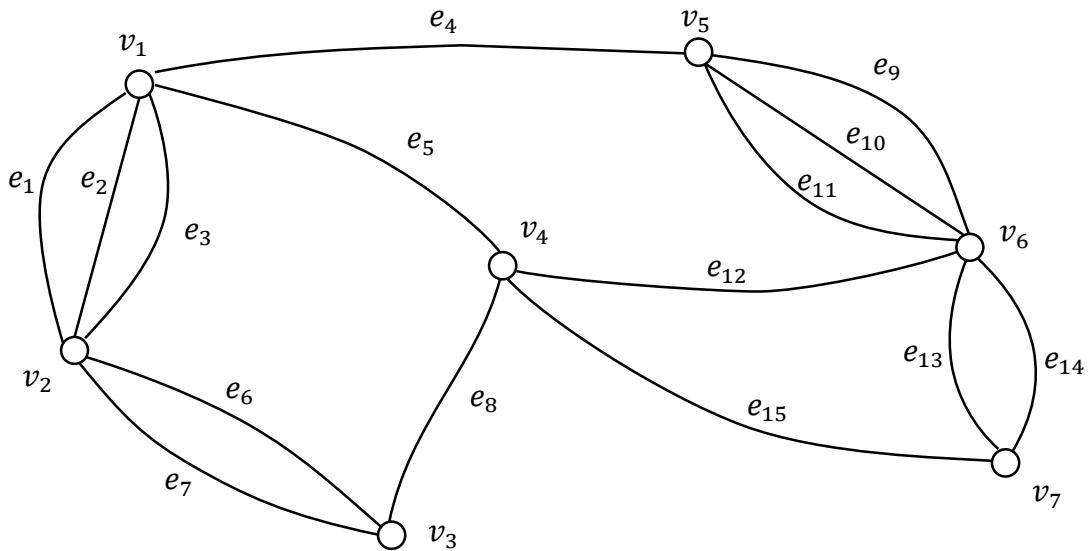


Рисунок 24 — Пример мультиграфа, отображающего маршруты общественного транспорта

Для осуществления пересадок все вершины дополнительно соединяются фиктивными ребрами, отражающими пешеходные маршруты. Для взвешивания ребер применяются два параметра:  $time(e_i)$  — время движения между остановками по ребру  $e_i$  и  $cost(e_i)$  — стоимость (стоимость билета) проезда по ребру  $e_i$ . В обоих случаях веса являются динамическими, что позволяет отобразить текущее состояние сети ОТ.

При планировании мультимодального маршрута критерий, связанный с минимальным расстоянием важен только в случае наличия пересадок, требующих от пользователя преодоления расстояния между соседними остановками пешком или с помощью велосипеда. В остальных случаях критериями для выбора кратчайшего пути могут быть минимальное время, затрачиваемое на него, минимальная стоимость, минимальное количество пересадок. Следует, однако, учитывать, что путь, удовлетворяющий только одному критерию, может иметь неприемлемые характеристики по другим, вследствие чего требуется проводить выбор пути с учетом нескольких критериев, решая задачу многокритериальной оптимизации. Обработка каждого критерия имеет особенности, описанные далее.

В связи с динамическим характером весов ребер, при планировании маршрута по полученному мультиграфу невозможно заранее оценить маршрут ни по одному из критериев, что не позволяет использовать алгоритм поиска кратчайших маршрутов  $A^*$ , требующий для работы эвристическую оценку кратчайшего расстояния. Единственным алгоритмом, позволяющим гарантированно получить результат в поставленных условиях, является алгоритм Дейкстры. Для поиска маршрута в мультиграфе с динамическими весами с учетом определенных выше

критериев была предложена модификация алгоритма Дейкстры: выбор вершин на каждом шаге осуществляется с учетом типа транспортного средства, номера маршрута и его расписания движения (листинг 1).

Модификация алгоритма, позволяющая учесть особенности формирования весов ребер мультиграфа, отображающего мультимодальную сеть, заключается в изменении принципа пометки вершин графа. Для каждой вершины меткой является момент времени, в который пользователь окажется в вершине и количество пересадок, требуемых для достижения вершины. Пересадкой считается выбор ребра, для которого  $rt \neq u.rt$ , где  $rt$  — номер нового маршрута,  $u.rt$  — номер маршрута, использованного для достижения вершины  $u$ .

**Параметры:**

$Graph(V, E, L)$  - мультиграф маршрутов  $OT$ ;

$rt(Num, Type, Cost) \in RT$  - маршрут  $OT$ . Включает в себя номер маршрута, тип транспорта, стоимость поездки;

$u(RT^* \in RT, T(RT^*)), v(RT^* \in RT, T(RT^*)) \in V$  - вершины графа = остановки, включают маршруты и их расписания;

$T(rt, v)$  - время прибытия маршрута  $rt$  на остановку  $v$  согласно расписанию;

$e(u, v, rt) \in E$  - ребро графа, задающее маршрут  $rt$  между вершинами  $u$  и  $v$

$time(e) \in L, cost(e) \in L$  - веса ребра  $e(u, v, rt)$  задающие время и стоимость пути соответственно

**Begin**

```

1. while u != end do //пока не достигнута конечная точка
2.   u = minT(rt) (u) //выбор вершины по минимальному
   времени
3.   for each neighbor v of u do // u,v - перебор соседних вершин к u
4.     for each e(u,v,rt) do // проверяются все маршруты между u,
   v
5.       if rt != u.rt do // маршрут не совпадает с
   минимальным в u
6.         alt_time = u.time + time(e) + (T(u.rt,u) - T(rt,u))
7.         alt_cost = u.cost + cost(e) //стоимость поездки
8.       else do // используется тот же маршрут
9.         alt_time = u.time + time(e)
10.        alt_cost = u.cost
11.        if alt_time < v.time do // найденный путь быстрее
12.          v.time = alt_time
13.          v.prev_time_stop = u
14.          v.cost = alt_cost
15.          v.rt = rt
End

```

Листинг 1 — Алгоритм Дейкстры для планирования мультимодального маршрута

Пересадка может осуществляться в текущей вершине или требовать перемещения между соседними вершинами с использованием другого типа транспорта. В первом случае время, требуемое на достижение следующей вершины, рассчитывается по имеющимся расписаниям и текущей дорожной ситуации по следующей формуле:

$$T(v, e_j) = T(u, e_i) + \text{time}(e_j) + (T(u, e_j) - T(u, e_i)), \quad (11)$$

где  $T(u, e_i)$  — время прибытия пользователя в вершину  $u$  по ребру  $e_i$ ;  $\text{time}(e_j)$  — расчетное время движения по ребру  $e_j$ ;  $T(u, e_j) - T(u, e_i)$  — время ожидания прибытия транспорта, следующего по ребру  $e_j$  с учетом текущего положения транспортного средства.

Во втором случае для перехода в новую вершину используется другой тип транспорта, что требует обращения к соответствующему сервису. При этом, если совершается пешеходный переход между остановками, что количество пересадок увеличивается на 1, а если с использованием другого типа транспорта, то на 2. В этом случае отсутствует время ожидания транспортного средства в новой вершине, но расчет времени достижения новой вершины осуществляется с учетом маршрута, полученного от другого сервиса:

$$T(v, e') = T(u, e_i) + \text{time}(e') \quad (12)$$

Вычислительная сложность модифицированного алгоритма складывается из вычислительной сложности алгоритма Дейкстры, которая увеличивается за счет наличия дополнительных ребер между вершинами и составляет  $O(|V| \log |V| \cdot |RT|)$ , где  $|V|$  — количество вершин мультиграфа  $|RT|$  — количество маршрутов между вершинами. По вершинам построенного пути `prev_time[v]` определяются типы используемых транспортных средств, номера маршрутов и точки их пересечения. В области точек пересечения маршрутов производится поиск остановок, между которыми осуществляется пересадка.

Планирование моно- и мультимодальных маршрутов может включать в себя планирование поездки с совместным использованием личного транспорта. Метод планирования данного типа маршрутов представлен в следующем разделе.

### **Планирование совместных поездок с использованием личного транспорта**

Планирование поездки с использованием личного транспорта предполагает выбор пользователей, владеющих автомобилями, и распределение между ними пользователей, запрашивающих вариант поездки между объектами на карте [7, 8]. Распределение осуществляется за счет поиска совпадающих путей между парами водитель-пассажир и последующего назначения водителя тому пассажиру, у которого найдется совпадение пути максимально удовлетворяют ограничениям, заданным пользователями.

На рисунке 25 представлена идея, лежащая в основе метода поиска совпадающего пути.

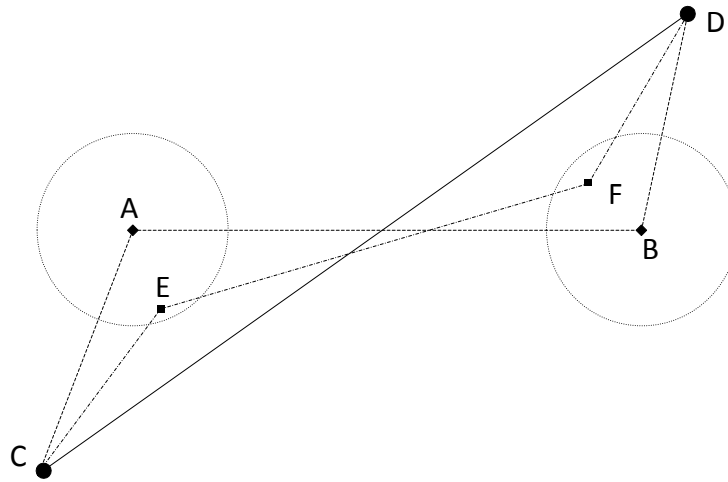


Рисунок 25 — Общий принцип поиска совпадающего пути

Пусть  $A, B$  — начальная и конечная точки пути пешехода,  $C, D$  — начальная и конечная точки пути водителя. Сплошная линия используется для обозначения самого короткого пути водителя, полученного с помощью ГИС. Начальные точки водителя и пассажира заданы так, чтобы общее направление движения водителя и пешехода совпадало. Идея совместной поездки заключается в корректировке пути водителя таким образом, чтобы он мог подвезти пешехода. Путь  $CABD$ , обозначенный на рисунке 25 пунктирной линией, является самым простым, так как точки встречи участников совместной поездки совпадают с точками начала и конца пути одного из них (пешехода). Возможны и более сложные варианты выбора пути, для которых поиск точек встречи, удовлетворяющих и водителя, и пассажира, производится в некотором радиусе от их ключевых точек, что требует повторного планирования путей всех участников совместной поездки. На рисунке 25 подобный вариант обозначен штрихпунктирной линией, соединяющей точки встречи и высадки, обозначенные буквами  $E, F$  (путь  $CEFD$ ). При выборе точек встречи и высадки должны учитываться следующие основные критерии:

- Суммарное расстояние от начальной точки пути пассажира до точки встречи и от точки высадки до конечной точки пути не должно превышать максимального пути, который согласен пройти пассажир (Detour). На рисунке 25 эта область обозначена пунктирной окружностью.
- Разность длины измененного пути водителя и самого короткого не должна превышать максимально допустимого отклонения, определенного в профиле водителя (Detour).

На рисунке 26 представлен алгоритм, по которому осуществляется выбор потенциальных точек встречи и высадки.

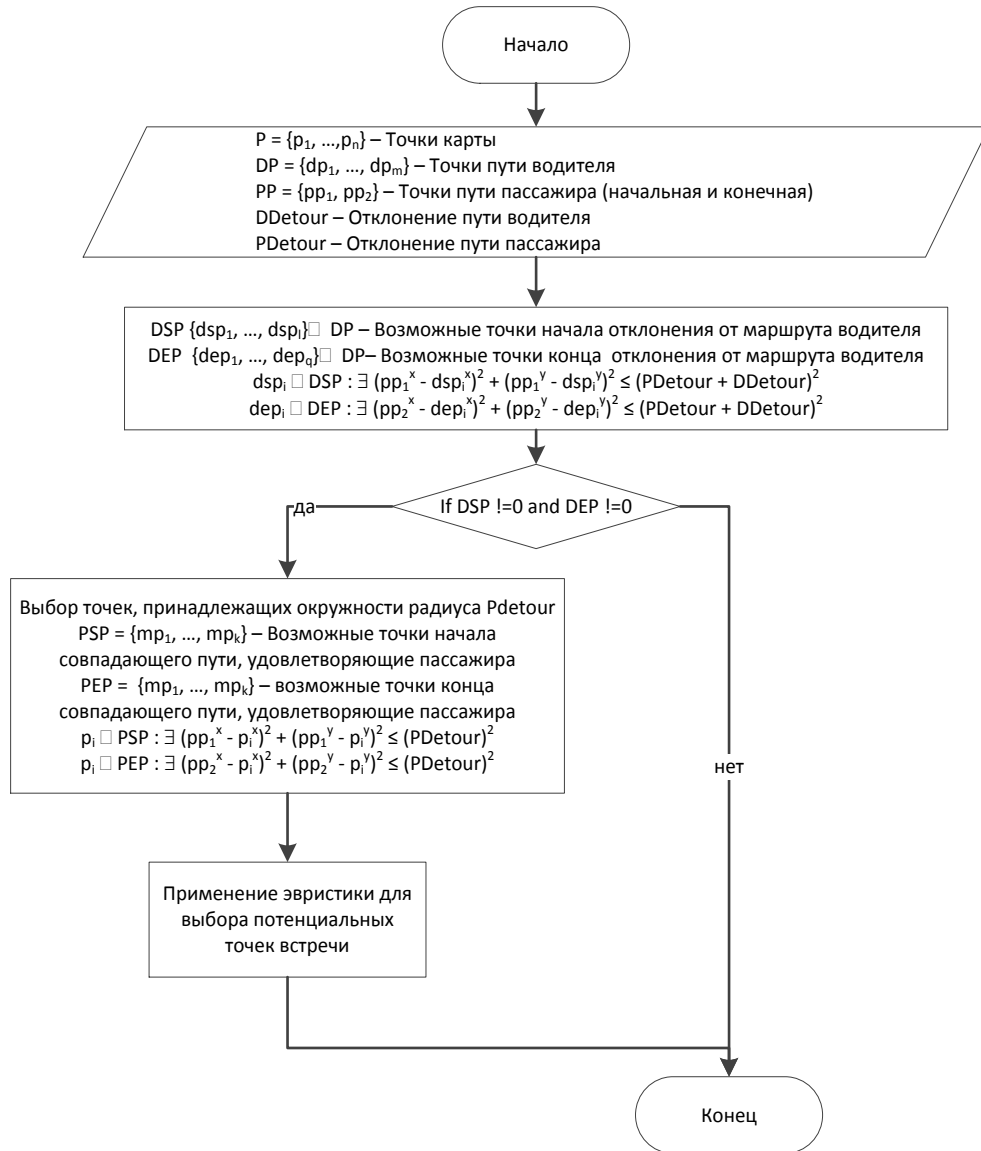


Рисунок 26 — Алгоритм выбора потенциальных точек встречи и высадки

Первым этапом работы алгоритма является определение принципиальной возможности подвоза пассажира выбранным водителем. Такая возможность существует при выполнении следующих условий:

$$(pp_1^x - dp_i^x)^2 + (pp_1^y - dp_i^y)^2 \leq (PDetour + DDetour)^2 \quad (13)$$

$$(pp_2^x - dp_i^x)^2 + (pp_2^y - dp_i^y)^2 \leq (PDetour + DDetour)^2, \quad (14)$$

где  $pp_1, pp_2$  — начальная и конечная точки пути пассажира,  $dp_i$  — точка пути водителя,  $PDetour, DDetour$  — отклонение от минимального пути пассажира и водителя. Если условия (13), (14) не выполняются, то у пассажира и водителя нет общих точек, удаленных на приемлемое для обоих расстояние. Таким образом водитель не может подвезти пассажира, что делает нецелесообразным выполнение следующих шагов алгоритма и его выполнение прекращается.



Если условия выполнены, то на следующем шаге составляется список точек пути водителя, соответствующих условиям, и осуществляется переход к следующему шагу, на котором производится выбор потенциальных точек встречи.

После выбора точек встречи для водителя, осуществляется выбор точек, до которых может дойти пешеход. Они принадлежат окружности радиуса  $PDetour$  и должны соответствовать следующему условию:

$$(pp_A^x - p_i^x)^2 + (pp_A^y - p_i^y)^2 \leq PDetour^2, \quad (15)$$

где  $pp_A$  — точка, принадлежащая пути пассажира,  $p_i$  — точка на карте. Если  $p_i$  удовлетворяет условию (3), то она выбирается как потенциальная точка встречи. На рисунке 27 точки, удовлетворяющие условию (3) обозначены буквами  $L, M, N, K$ .

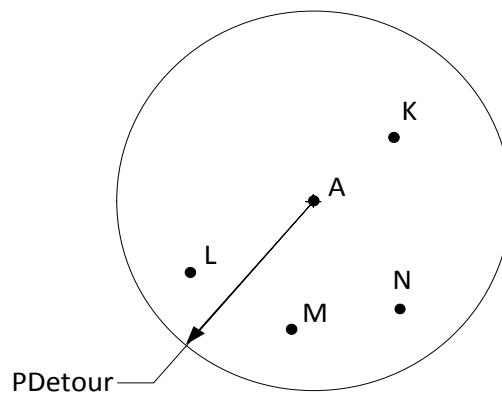


Рисунок 27 — Выбор точек встречи для пассажира

Для выбора из всех потенциальных точек пары, наиболее удовлетворяющей критериям, заданным пассажирами, осуществляется перебор всех комбинаций точек встречи и точек высадки с вычислением требуемых критериев для выбора маршрута. Вычисление всех возможных комбинаций путей является крайне затратной операцией в силу многократного запуска функций поиска кратчайшего пути на графе дорог. Одним из возможных способов снижения затрат является сокращение множества потенциальных точек встречи. Для этого были разработаны эвристики, позволяющие на основе имеющихся данных произвести отсев потенциальных точек встречи и высадки, устранив точки, предположительно дающие худшие комбинации.

Работа первой эвристики предполагает определение сектора окружности, со стороны которого будет ехать водитель и выбор точек из него. На рисунке 28 представлена эвристическая модель для ситуации, в которой у водителя нашлась только одна точка (точка  $C$ ), удовлетворяющая одному из условий (13) или (14).

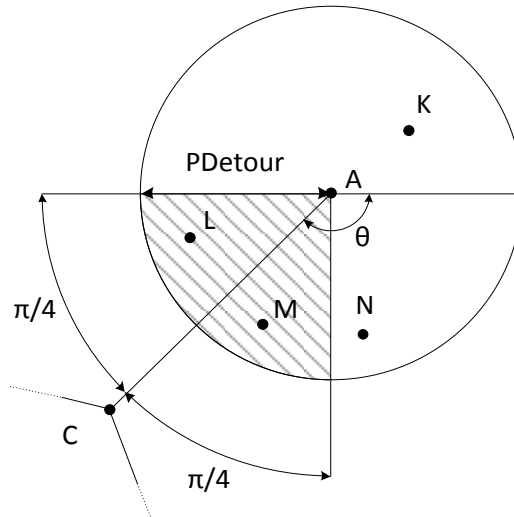


Рисунок 28 — Схема для одной точки пути водителя

Для определения потенциальных точек встречи и высадки вычисляется угол  $\theta$ :

$$\theta = \text{arctg} \left( \frac{C^y - A^y}{C^x - A^x} \right) \quad (16)$$

и выбираются точки, принадлежащие области  $\left[ \theta - \frac{\pi}{4}, \theta + \frac{\pi}{4} \right]$ . На рисунке 28 в область, ограниченную этими углами, попадают точки L и M. За счет того, что рассматриваются точки, попадающие в сектор окружности, точка A, принадлежащая пассажиру, всегда будет входить в множество потенциальных точек встречи или высадки. Если нашлись две и более точки, удовлетворяющих условиям (13) или (14), то область выбора точек расширяется. На рисунке 29 представлена схема применения эвристики, полученной из схемы, изображенной на рисунке 28, представляющая пример ситуации с обнаружением большего количества точек водителя, удовлетворяющих условию (13) или (14). Расширенная область, помимо точек L и M уже включенных в список потенциальных точек встречи или высадки, содержит точку N.

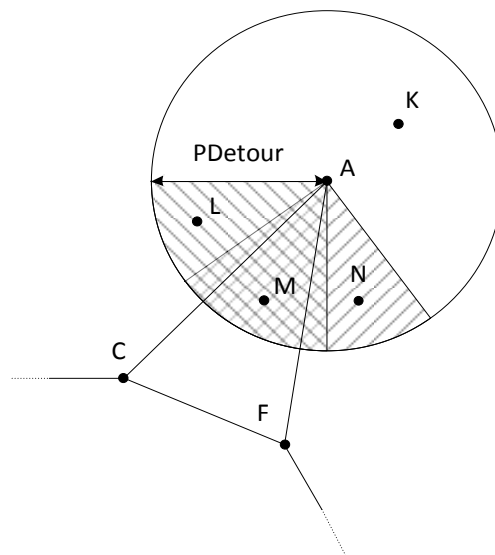


Рисунок 29 — Схема анализа двух точек пути водителя

Применение эвристики, основанной на выборе точек в секторе круга, ограниченного дальностью пешеходного маршрута пассажира, позволяет рассматривать только точки, находящиеся со стороны движения водителя. Однако, она обладает существенными недостатками:

- Могут быть выбраны точки, расположенные дальше, чем максимальное отклонение водителя от кратчайшего пути.
- Искусственно ограниченная область выбора. Некорректный выбор угла может привести к потере потенциальных точек встречи.

Вторая эвристика учитывает недостатки, выявленные при анализе первой. Схема применения второй эвристики представлена на рисунке 30.

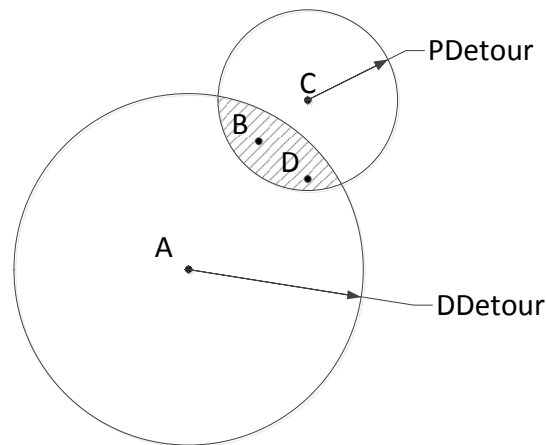


Рисунок 30 — Схема применения второй эвристики

Суть второй эвристики заключается в том, что выбор точек осуществляется на пересечении окружностей радиусов  $PDetour$  и  $DDetour$ , с центрами в точках, принадлежащих пассажиру и водителю соответственно. За счет этого все точки, находящиеся в области пересечения окружностей, являются достижимыми как для водителя, так и для пассажира, при этом отпадает необходимость определения угла, ограничивающего область выбора. Пересечение окружностей также может быть расширено при построении окружностей по соседним точкам пути водителя, удовлетворяющих условию (13) или (14), и объединении пересекающихся областей, как это описано в первой эвристике.

Эффективная работа эвристик требует соблюдения следующих ограничений:

- Большое количество водителей и пассажиров. В целом данное ограничение справедливо к самой идее совместных поездок с использованием личного транспорта. При малом количестве пользователей, из-за строгости эвристик количество потенциальных точек встречи будет достаточно малым и поиск совпадающих маршрутов будет редко давать положительный результат.

- Малое значение минимального отклонения водителя от самого короткого маршрута (*Detour*). При больших значениях (*Detour*) применение эвристики будет бесполезным.
- Дороги на карте должны быть распределены равномерно. Неравномерность (наличие рек, озёр, заводов и т.п.) ведет к отсутствию дорог в некоторых секторах, что может привести к потере возможных точек встречи, связанной с необходимостью объезда препятствия и заезда за пешеходом с другой стороны при больших значениях *DDetour*.

Среди точек, оставшихся после применения эвристик, осуществляется расчет параметров нового пути для всех комбинаций точек встречи и точек высадки (функция `common_shortest_paths`). Этими параметрами являются новая длина пути водителя, расстояние, преодолеваемое пассажиром до точки встречи, время ожидания в точке встречи для водителя и для пассажира.

**Параметры:**

DR – список водителей; PS – список пассажиров;  
 dr[s], dr[e] – начальная и конечная точки пути водителя;  
 ps[s], ps[e] – начальная и конечная точки пути пассажира;  
 MP[s], MP[e] – потенциальные точки встречи и высадки;

**Begin**

```

1. For each dr ∈ DR do
2.   dr[path] = dijkstra_shortest_path(dr[s], dr[e]) // путь водителя
3.   For each ps ∈ PS do
4.     MP[s]=find_meeting_points(dr[path], ps[s])// потенциальные точки
встречи
5.     MP[e]=find_meeting_points(dr[path], ps[e])// потенциальные точки
высадки
6.   For each start_MP ∈ MP[s] do
7.     For each end_MP ∈ MP[e] do
8.       Common_path=common_shortest_paths(
                               dr[path],ps[s],ps[e],start_MP,end_MP)
9.       If check_preferences(dr, common_path) and
10.        check_preferences(ps, common_path) then // проверка
условий
11.         dr[fellow] = ps // назначение попутчика водителю
12.         ps[fellow] = dr // назначение попутчика пассажиру
End

```

Листинг 2 — Алгоритм поиска попутчиков при совместном использовании автотранспорта

Расчет нового пути для водителя начинается с проверки совпадения выбранной точки встречи с одной из точек, принадлежащих текущему пути водителя. При отсутствии совпадения вычисляется положение выбранной точки встречи относительно существующих точек пути водителя. Для этого выбираются две соседние точки существующего пути и производится расчет

расстояния от новой точки до выбранных. Положение новой точки определяется парой, дающей наименьшее расстояние. На рисунке 31 представлен вариант пересчета пути водителя, в котором точка встречи  $F$  расположена между существующими точками пути водителя  $C_i$  и  $C_{i+1}$ , соответствующими точкам встречи с пассажирами, которых выбранный водитель к данному моменту времени уже согласился подвезти. Точка  $A$  на рисунке обозначает точку пути пассажира.

После определения положения точек встречи и высадки необходимо проверить наличие свободных мест в транспорте водителя на отрезке между ними. При отсутствии свободных мест работа алгоритма прекращается. Если свободные места есть, то производится расчет новой длины пути водителя, расстояния, преодолеваемого пассажиром до точки встречи и времени ожидания водителя и пассажира.

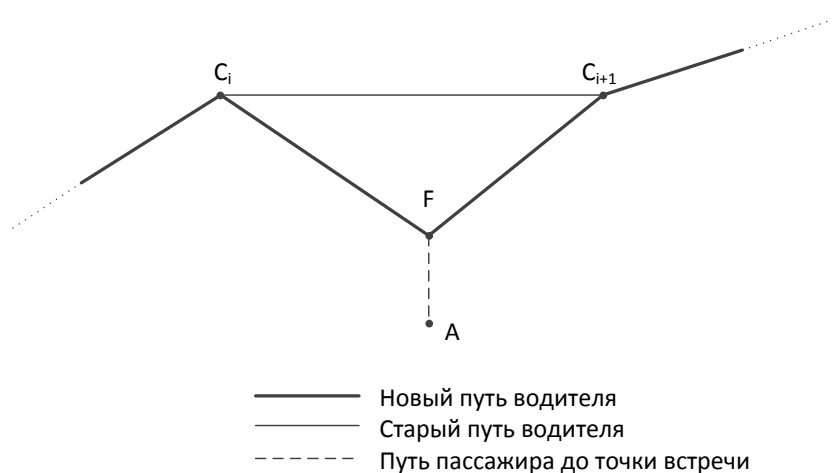


Рисунок 31 — Добавление новой точки к пути водителя

Также, для всех точек пути пересчитываются временные метки с учетом времени ожидания и времени, затрачиваемого на передвижение до точек встречи. По результатам этих расчетов будет принято решение о выборе наилучшего пути. Критерии, по которым осуществляется принятие решения, определяются пользователями при заполнении своего профиля и размещении его в интеллектуальном пространстве. Этими критериями могут быть:

- Минимальная длина пути водителя (Detour);
- Минимальное время ожидания в точке встречи (Delay);
- Минимальное расстояние до точек встречи (актуально для пассажира, Detour).

### 3.4 Метод построения сервиса рекомендаций объектов

#### Модели формирования рекомендаций

Необходимость в формировании рекомендаций в системе обеспечения инфомобильности связана с большим объемом информации, предоставляемой пользователю в результате работы

отдельных сервисов. Из всего объема информации только часть может быть действительно интересна и полезна пользователю. Для повышения качества предоставляемой информации список объектов, найденных сервисом поиска объектов, и информация о них должна быть отсортирована с учетом различных критериев. Для этой цели в систему обеспечения инфомобильности включен сервис рекомендаций, основная цель работы которого — сбор информации об интересах пользователей и выработка рекомендаций для конкретных пользователей, на основе которой будет отсортирована вся предоставляемая информация.

В зависимости от типа информации, используемой для формирования рекомендаций, выделяют шесть классов рекомендуемых систем [10]:

- 1) Контентные. Используют формализацию представления объектов для определения близости рекомендуемых объектов. При использовании подобных систем необходима строгая формализация и типизация объектов.
- 2) Коллаборативной фильтрации. Наиболее распространены в силу простоты реализации. Рекомендации формируются только на основе оценок, присвоенных объектам.
- 3) Демографические. Основываются на демографической информации о пользователе, в связи с чем применяются в основном в социологии и маркетинге.
- 4) Основанные на знаниях. Требуют формализации того, как характеристики объектов сочетаются с задачами пользователя. При выработке рекомендации используют вывод на основе прецедентов и удовлетворение ограничений.
- 5) Социальные. При выработке рекомендации основываются на информации об отношениях между пользователями, близости их вкусов и т.д.
- 6) Гибридные. При формировании рекомендаций используют свойства и алгоритмы всех вышеперечисленных классов систем.

При выработке рекомендаций объектов в системе обеспечения инфомобильности для конкретного пользователя учитываются предыдущие оценки объектов, выставленные как в системе обеспечения инфомобильности, так и в других сервисах. Таким образом, при разработке сервиса рекомендаций используется подход коллаборативной фильтрации [10], который основывается на оценке сходства между парами пользователей или парами объектов. Так как одним из требований системы обеспечения инфомобильности является персональный характер поддержки, то основным подходом при выработке рекомендаций является подход, использующий отношение между пользователями (user-based).

При выработке рекомендации необходимо дополнительно учитывать контекст пользователя для предложения объектов, наиболее релевантных в сложившейся ситуации. Так,

при поиске кафе, в зависимости от погоды может быть предложено кафе под открытым небом в случае солнечного дня или любое другое кафе в случае дождя. В связи с этим, сервис рекомендаций также должен обеспечить сбор контекста при выставлении пользователем оценки, ассоциацию оценки с контекстом и вычисление новой оценки с учетом текущей ситуации [16].

### **Сбор и обобщение контекста для выработки рекомендации**

В системе обеспечения инфомобильности при формировании рекомендаций предлагается использовать следующие контекстные характеристики:

- 1) Дата, время;
- 2) Компания, в которой находится пользователь: один, с друзьями или с семьей;
- 3) Погода. Текущие погодные условия: температура, влажность, осадки и т.п.

Данные классы составляют контекст пользователя и конкретные атрибуты могут быть извлечены из интеллектуального пространства напрямую, либо вычислены косвенно. Так, для каждого пользователя определяется его время и текущее местоположение. Если погода не была определена заранее, то она может быть запрошена у погодного сервиса. Компания, в которой находится пользователь также может быть определена косвенно за счет анализа местоположения родственников и друзей, полученных из аккаунтов в социальной сети. Также, подобная информация может быть введена пользователем напрямую.

Существует три основных подхода, позволяющих использовать контекст при формировании рекомендаций [32; 33]:

- 1) Предварительная обработка контекста. Информация о текущем контексте используется для выбора или формирования релевантного множества оценок. После этого конкретные оценки могут быть предсказаны с использованием традиционных способов формирования рекомендаций.
- 2) Постобработка контекста. Оценки формируются с использованием традиционных техник по всем имеющимся данным. И только после формирования набора оценок они корректируются для каждого пользователя с использованием контекстной информации.
- 3) Моделирование контекста. Контекст используется в момент формирования рекомендации.

При формировании рекомендаций в соответствующем сервисе системы обеспечения инфомобильности используется предварительная контекстная обработка. Ее выбор обусловлен возможностью применения данной техники совместно с алгоритмами коллаборативной фильтрации и возможностью обработки оценок, релевантных текущей ситуации. Следует,

однако учитывать, что предварительная обработка контекста может привести к проблемам, связанным с недостаточным объемом выборки оценок, требуемых для корректного формирования рекомендаций.

Несмотря на то, что атрибуты контекста ограничены соответствующими доменами, значения атрибутов могут варьироваться в достаточно широких пределах. Для выделения общих характеристик атрибутов применяется метод обобщения контекста, разработанный Adomavicius G. и др., в 2005 году [34]. Данный метод позволяет фильтровать матрицу рейтингов не только по конкретным значениям атрибутов контекста, но и по их обобщениям, что позволяет избежать проблемы недостаточности выборки. Для использования данного метода модель контекста должна содержать правила обобщения контекста.

Процесс обобщения атрибутов класса контекста представляет собой отображение значений атрибутов из текущего домена класса в значения более общего домена, определенного при разработке сервиса рекомендаций. Формальное представление, описанное в работах [32 – 34], было адаптировано для применения с разработанной моделью контекста:

$$f(d): D_{op} \rightarrow D'_{op} \quad (17)$$

где  $f(d)$  — функция обобщения;  $\rightarrow$  — отношение обобщения, задающее отображение элементов множества значений атрибутов контекста во множество его обобщений;  $D'_{op}$  — домен более высокого уровня, содержащий обобщения значений домена  $D_{op}$ . В большинстве случаев, при последовательном применении обобщений, такое отношение формирует иерархию отображений (см. рисунок 32).

Обобщение контекста для выбранных атрибутов осуществляется следующим образом:

- 1) Дата и время могут быть обобщены разными способами, среди которых выделяется, в зависимости от степени общности, время суток (утро, день, вечер, ночь), день недели (понедельник – воскресенье), тип дня недели (будний или выходной день), месяц и время года (зима, весна, лето, осень). Самым большим обобщением считается значение «любое время»;
- 2) Компания. При обобщении находящихся рядом людей, используются категории «один», «семья», «друзья». Самым большим обобщением данной характеристика является «любая компания»;
- 3) При обобщении погодных условий применяются категории «солнечно», «дождливо», «облачно», «снежно». Здесь самым большим обобщением является «любая погода».



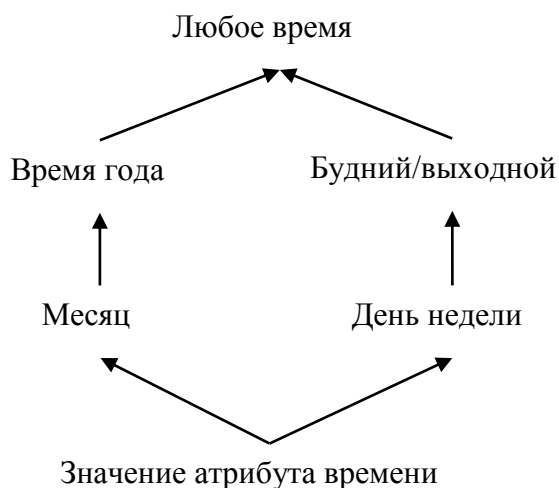


Рисунок 32 — Пример построения иерархии при обобщении контекста

В качестве примера может служить следующая ситуация, в которой контекст, заданный атрибутами (Время: 28.09.2015 17:30 MSK, Окружение: («Петр Иванов», связь: «друг»; «Сергей Васильев», связь: «друг»)) Погода: 10°C, 750 мм рт. ст., «дождь») преобразуется в следующий вид: (время: «осень», компания: «друзья», Погода «дождливо»). Следует отметить, что контекст может обобщаться различными способами, в зависимости от используемого домена. Чем больше атрибутов учитывается при обработке контекста, тем больше возможных способов обобщения существует. Для выбора способа, наиболее подходящего целям решаемой задачи, разработаны различные эвристики [34].

В рамках задачи, стоящей перед системой обеспечения инфомобильности, достаточно учитывать представленные выше атрибуты, в связи с чем, использования дополнительных эвристик для выбора способа обобщения контекста не требуется. Извлечение оценок с учетом контекста и формирование рекомендации описаны в следующем разделе.

### **Формирование рекомендаций, использующих отношение близости между пользователями и учитывающих контекст**

Для каждого объекта, предложенного системой обеспечения инфомобильности, может быть выставлена оценка по пятибалльной шкале. Оценен может быть как сам объект, так и фотографии, подобранные для него, и его описание. Текущий контекст сохраняется вместе с выставленной оценкой.

Для снижения субъективности, неизменно возникающей при сборе большого количества оценок от пользователей, используется нормализация оценок, выставленных конкретным пользователем. Нормализация осуществляется с использованием следующей формулы [161]:

$$r'_{ui} = r_{ui} - \frac{1}{|K_u| + 1} \left( 3 + \sum_{k \in K_u} r_{uk} \right), \quad (18)$$

Где  $r_{ui}$  — необработанная оценка объекта  $i$ , выставленная пользователем  $u$ ,  $K_u$  — набор объектов, оцененных пользователем  $u$ . Смысл нормализации заключается в переходе от пятибалльной шкалы, используемой пользователями к пригодной для расчетов шкале, выстроенной относительно среднего значения выставленной пользователем оценки. Переход обусловлен тем, что пользователи при выставлении оценки предпочитают использовать часть предоставленного диапазона. Из-за этого, при использовании пятибалльной шкалы большая часть оценок будет сосредоточена либо в конце диапазона («3»–«5»), либо в его начале («1»–«3»). При последующем вычислении рекомендации подобный разброс может оказать негативное влияние на предполагаемую оценку, которая может быть либо завышена, если чаще всего выбирается верхний диапазон, либо занижена при выборе нижнего. Нормализация шкалы относительно средней оценки позволяет решить данную проблему. Следует учитывать, что на первых этапах работы, когда количество оценок мало, использование нормализации становится невозможным, в связи с чем необходимо введение временного фиктивного среднего значения. При использовании пятибалльной шкалы таким значением является «3». При накоплении достаточного количества оценок влияние фиктивной оценки на общий результат нивелируется.

Задача формирования рекомендации заключается в предсказании оценки, которая с наибольшей вероятностью будет присвоена пользователем выбранному объекту. При вычислении предположительной оценки в первую очередь рассматриваются те оценки, контекст которых совпадает с текущим контекстом пользователя. Для этого последовательно сравниваются сначала конкретные контексты, а затем различные обобщения текущего контекста пользователя и контекст оценки.

Вычисление оценки, после выбора оценок, соответствующих контексту, выполняется в два этапа. На первом определяются группы пользователей, имеющих оценки сопоставимые с оценками пользователя, для которого формируется рекомендация. Группы формируются на основе метода  $k$  ближайших соседей (*k-nearest neighbors algorithm*,  $k$ -NN) Для вычисления степени сходства используется косинусная мера, представленная в формуле (19) [161].

$$s_{uv} = \frac{\sum_{o \in O} r'_{uo} r'_{vo}}{\sqrt{\sum_{o \in O} r'^2_{uo} \sum_{o \in O} r'^2_{vo}}} \quad (19)$$

где  $u, v$  — пользователи;  $O$  — множество оцененных объектов;  $r'_{uo}, r'_{vo}$  — нормализованные оценки объектов для пользователей;  $s_{uv}$  — степень сходства оценок пользователей. После

формирования групп вычисляется вероятная оценка на основе оценок, поставленных пользователями из соответствующей группы для соответствующего объекта.

$$r_{uj}^* = \frac{\sum_{v \in G} r'_{vj} s_{uv}}{\sum_{v \in G} |s_{uv}|}, \quad (20)$$

Где  $r_{uj}^*$  — предполагаемая оценка пользователя  $u$ ;  $G$  — выбранная группа пользователей со сходными оценками.

Подобным образом вычисляются предполагаемые оценки для всех объектов, предлагаемых пользователю. При итоговом ранжировании объектов дополнительно учитывается, насколько важным для пользователя является удаленность объекта от текущего местоположения [161].

$$s_j = kr_{uj}^* + (1 - k) \left( 1 - \frac{d_j^w}{\max_{i \in L} d_i^w} \right), \quad (21)$$

Где  $s_j$  — итоговая оценка объекта  $j$ ;  $k \in [0,1]$  — параметр, определяющий какой критерий важнее для пользователя: оценка объекта или его удаленность от текущего местоположения;  $d_j^w$  — оценка времени, требуемого для достижения объекта;  $L$  — список объектов. Ранжирование объектов осуществляется по убыванию итоговой оценки  $s_j$ .

### 3.5 Модель построения сервиса обеспечения конфиденциальности информации пользователей

#### Модели обеспечения конфиденциальности в интеллектуальных системах

Для обеспечения персонализации результатов работы сервисов системы обеспечения инфомобильности требуется сбор и обработка контекста, который включает в себя в том числе и информацию о пользователе [25]. Часть этой информации может быть обезличена перед передачей и обработкой, например, информация о дорожной ситуации в местоположении пользователя. Информация, которая не может быть обезличена, без обеспечения разделения доступа к ней позволяет провести ассоциацию данной информации с определенным пользователем, что может повлечь за собой несанкционированный сбор, обработку и накопление информации о пользователе, позволяющей идентифицировать его личность. Таким образом, в системе обеспечения инфомобильности требуется обеспечение конфиденциальности информации о пользователе. Конфиденциальность информации — «обязательное для выполнения лицом, получившим доступ к определенной информации, требование не передавать такую информацию третьим лицам без согласия ее обладателя» [20].

Основной проблемой современного развития интеллектуальных пространств является отсутствие проработанных подходов к управлению доступом к ресурсам. Причем при

разграничении доступа должна учитываться текущая ситуация (например, доступ к файловому серверу компании из локальной сети и «из вне» её может обрабатываться по-разному). В связи с этим проектирование моделей и разработка подхода, направленного на обеспечение контекстно-зависимого доступа к ресурсам интеллектуального пространства является на сегодняшний день актуальной и востребованной задачей, что также подтверждается большим количеством исследований в данной области за последние три года [11].

В работе [35] предлагается подход, который объединяет контекстные знания с процессом автоматизированного логического вывода для осуществления аутентификации и контроля доступа в вычислительных системах, основанных на интеллектуальных пространствах. Этот подход авторы используют в ядре проекта Gaia, который предоставляет инфраструктуру для построения интеллектуальных пространств. Контроль доступа основан на вычислении уровня доверия пользователю. Этот уровень вычисляется на основе контекста пользователя с использованием вероятностей, Байесовской вероятности и нечеткой логики. Каждое значение уровня ассоциируется с разной силой аутентификации, которая позволяет совершать различные действия в интеллектуальном пространстве. Подобный подход является довольно гибким, и применим в динамических системах, какими являются интеллектуальные пространства.

В статье [105] предлагается объединение разграничения доступа на основе ролей и атрибутов. Рассматривается три варианта объединения: с динамическими ролями, в которых атрибуты задают роли пользователей в системе; атрибутивное, в котором роль это атрибут (а не набор разрешений); ролевое, в котором атрибуты добавляются к ограниченным правилам ролевого контроля доступа, при этом они могут только уменьшать количество допустимых разрешений для пользователя, но не увеличивать их. За счет объединения ролей с атрибутами появляется возможность предоставлять доступ в зависимости от текущей ситуации (контекста), например, от времени суток или территориального расположения пользователя.

Развивая похожие идеи, работа [118] объединяет свойства и идеи моделей на основе ролей и атрибутов и предлагают онтологическую модель разграничения доступа. Использование онтологии обеспечивает принятие решения по уровню доступа к информации и предоставляет возможность автоматического поиска информации о контроле доступа.

В работах [55] и [56] предлагается система для контроля доступа в социальных сетях, основанную на технологиях семантической паутины. Реализация, представленная в статье, позволяет поддерживать контроль доступа не только за счет "прямой дружбы" с владельцем ресурса, но и оценивать степень доверия владельца пользователю и в зависимости от этой оценки предоставлять доступ. Предлагается политика фильтрации доступных ресурсов, задаваемая правилами, как и политика контроля доступа. Благодаря этим политикам, лицо,

предоставляющее доступ, может контролировать информацию, предоставляемую целевому пользователю.

Технологии семантической паутины используется также в работе [90]. Контроль доступа строится на основе модели ролевого контроля с использованием некоторых идей атрибутивного контроля, а именно, расширением атрибутами, удостоверяющими личность (сертификаты X.509 [61], публичный ключ и др.). Авторы предлагают архитектуру системы, на основе описанной модели, и ее реализацию.

В работе [174] проводится сравнительный анализ двух моделей контроля доступа: на основе ролей и на основе атрибутов, применительно к семантической паутине. Для каждой модели авторы приводят краткое описание и выделяют сильные и слабые стороны. Одним из плюсов атрибутивного контроля над ролевым, авторы выделяют поддержку контекста за счет атрибутов, что позволяет учитывать текущую ситуацию при выдаче прав доступа.

В работе [179] предлагается модель безопасности для интеллектуальных пространств, построенных на платформе Smart-M3 [92]. Автор рассматривает алгоритмы идентификации, аутентификации и контроля доступа. Для авторизации в Smart-M3 предлагается использовать HIP (Host Identity Protocol) [121], а в качестве модели контроля доступа — дискреционную модель, с матрицей, содержащей все разрешения для пользователей. Для работы данной модели авторы предлагают делать отображение интеллектуального пространства на виртуальную файловую систему, в которой каждый элемент пространства представлен отдельным файлом, а их иерархия — вложенными папками. Модуль, реализующий контроль доступа, автор встраивает в платформу Smart-M3.

Все упомянутые работы, за исключением [35], имеют своей целью адаптацию уже существующих моделей разграничения доступа к особенностям технологии семантической паутины и, следовательно, применимы к интеллектуальным пространствам. Вследствие того, что концепция интеллектуального пространства объединяет в себе идеи распределенных сетей и семантической паутины, модель разграничения доступа должна обеспечивать интероперабельность, гибкость и простоту правил разграничения доступа, децентрализованность предоставляемого ресурса и возможность принятия решения на основе семантических атрибутов, составляющих контекст. Из всех представленных моделей наиболее полную поддержку представленных выше условий осуществляют модели на основе атрибутов и онтологические модели, строящиеся на базе комбинации ролевого и атрибутивного разделения доступа, а также схема, предложенная в работе [35]. Модель, предложенная в работе [179], имеет ряд ограничений, напрямую связанных с ограничениями дискреционной модели разделения доступа, такими как статичность, отсутствие поддержки контекста и сложность конфигурирования. Отображение

ресурсов интеллектуального пространства в виртуальную файловую систему требует затрат вычислительных мощностей, что непременно скажется на быстродействии системы.

В следующем разделе предлагается метод обеспечения конфиденциальности информации о пользователе, учитывающий особенности взаимодействия сервисов в интеллектуальном пространстве на базе платформы Smart-M3, и позволяющий обеспечить доступ к информации на основе контекста сервиса, в котором происходит доступ.

### **Метод построения сервиса обеспечения конфиденциальности информации в интеллектуальном пространстве**

Для обеспечения конфиденциальности информации о пользователе предлагается выделить в интеллектуальном пространстве защищенную (приватную) область, обращение к которой могут осуществлять только две стороны: сервис, предоставляющий информацию и сервис, ее запрашивающий. Решение о предоставлении информации запрашивающему сервису принимается на основе его контекста, который должен соответствовать шаблону, заданному сервисом, предоставляющим информацию. Модель организации взаимодействия с приватной частью интеллектуального пространства представлена на рисунке 33.

Приватная область интеллектуального пространства представляет собой зашифрованные RDF-тройки в публичном интеллектуальном пространстве. Шифрование осуществляется с помощью алгоритма асимметричного шифрования RSA. Выбор асимметричного способа шифрования основывается на возможности использования разных ключей для операций шифрования и дешифрования, и обеспечения достаточной стойкости алгоритма к взлому в рамках поставленной в работе задачи. Разделение ключей позволяет держать в открытом доступе только ключ, по которому проводится шифрование, таким образом сервис, которому требуется передать информацию в приватном режиме использует открытый ключ сервиса, которому надлежит передать информацию, для проведения операции шифрования, результат которой размещается в открытом интеллектуальном пространстве. Сервис, ключом которого было проведено шифрование получает соответствующие тройки и может их расшифровать с помощью своего закрытого ключа.

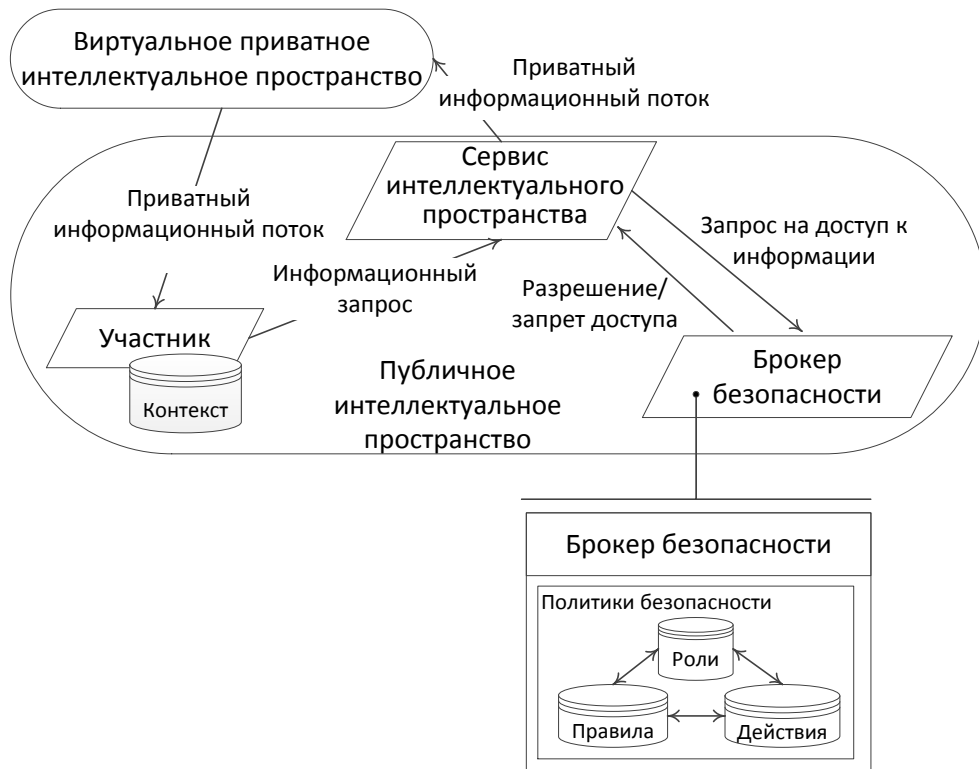


Рисунок 33 — Модель контекстно-зависимого разделения доступа к ресурсам интеллектуального пространства

Для обеспечения возможности работы с приватным интеллектуальным пространством требуется изменение порядка взаимодействия сервисов для запроса информации. При использовании «классной доски», на основе которой построено интеллектуальное пространство, информация, размещенная в пространстве, является общедоступной и для ее получения сервисам достаточно сформировать прямой запрос к семантическому информационному брокеру. При выделении части информации в приватную область сервисам перед формированием запроса, требуется получить доступ к этой области. Доступ предоставляется после того, как специализированный сервис, названный брокером безопасности, проанализирует контекст сервиса, запрашивающего доступ к информации на его соответствие разрешенным параметрам для запрашиваемого типа информации. Порядок взаимодействия сервисов посредством виртуального приватного интеллектуального пространства представлен на рисунке 34.

Пусть «пользователь 1» — сервис, которому требуется конфиденциальная информация от сервиса, представляющего пользователя — «пользователь 2». «Пользователь 1» формирует и отправляет в публичное ИП запрос, содержащий тип информации, к которой требуется доступ. «Пользователь 2» получает этот запрос и формирует следующий запрос, относительно того, имеет ли «пользователь 1» доступ к запрашиваемой информации, который отправляется на обработку брокеру безопасности. Брокер безопасности собирает контекст «пользователя 1» и проверяет его на соответствие критериям доступа к запрашиваемому типу информации, согласно политиками безопасности, представленным далее. По результатам проверки брокер

безопасности отправляет «пользователю 2» либо разрешение на передачу информации, либо запрет на соответствующие действия.

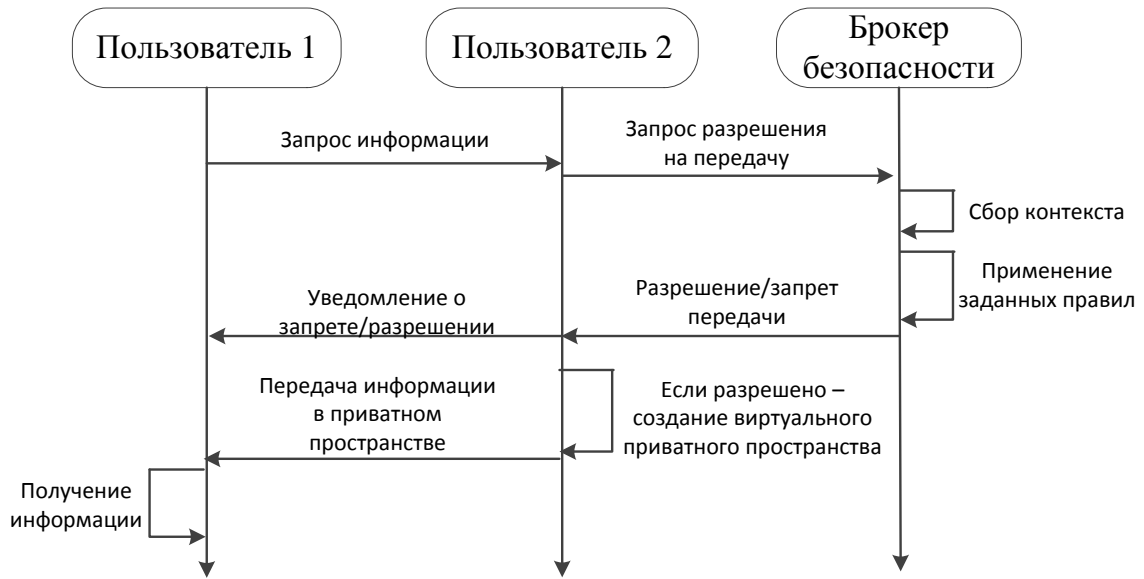


Рисунок 34 — Порядок взаимодействия посредством использования виртуального приватного интеллектуального пространства

При получении разрешения на передачу «пользователь 2» запрашивает из публичного ИП открытый ключ «пользователя 1» и шифрует требуемую информацию с использованием этого ключа. Вместе с тем, с помощью алгоритма SHA-1 [70] вычисляется контрольная сумма от передаваемой информации, которая шифруется с помощью закрытого ключа «пользователя 2», формируя цифровую подпись к сообщению, которая используется для подтверждения источника информации. Зашифрованные RDF-тройки вместе с подписью размещаются в публичном ИП, формируя, тем самым, виртуальное частное ИП. «Пользователь 1» может получить зашифрованные тройки и расшифровать их с помощью своего закрытого ключа, получив, тем самым, требуемую информацию. С помощью цифровой подписи он также может проверить действительно ли информация получена от «пользователя 2».

Результат обработки пересылается от «пользователя 1» к «пользователю 2» с помощью того же порядка действий, за исключением обращения к брокеру безопасности. Для шифрования используется открытый ключ «пользователя 2», а для формирования цифровой подписи — закрытый ключ «пользователя 1». Для исключения возможности подмены брокера и ответов от брокера взаимодействие с ним также проходит с помощью создания цифровых подписей для RDF-троек.

Для принятия решения о предоставлении доступа сервису используется его контекст. На основе контекста участника определяется уровень доверия к нему, на основе которого задается роль в системе. Разделение на роли позволяет облегчить политики безопасности и сделать их понятными для человека, конфигурирующего брокер безопасности. Значение каждого атрибута



класса контекста ассоциировано с определенным уровнем доверия, который представлен числом в интервале  $[0, 1]$ . Конкретное значение каждой из составляющих зависит от контекста текущей ситуации. Например, уровень доверия “0.2” и “0.9” может быть присвоен для доступа из публичной и частной сети соответственно. Для задания роли участнику используется логическая функция произведения, операндами которой являются логические значения «Правда» и «Ложь», полученные при вычислении соответствия текущего значения уровня доверия и требуемого для получения определенной роли. Например, роль «доверенный\_участник» может быть присвоена тогда и только тогда, когда пользователь аутентифицирован, используется частная сеть (уровень доверия к сети лежит в промежутке  $[0.8, 1]$ ), и текущее время не позднее 17:00 (уровень доверия по времени лежит в промежутке  $[0.3, 1]$ ). В соответствии с представленным описанием, в политиках безопасности можно выделить три типа правил:

- 1) Первый тип правил используется для задания соответствия семантического значения компоненты контекста ее цифровому представлению в брокере безопасности. Для каждого класса  $o_i$  определена соответствующая функция:

$$f_{trust_i}(o_i, q_i) \rightarrow [0, 1] \quad (22)$$

Примеры этого типа правил следующие:

```
TrustValue(public_network) = 0.1;
TrustValue("08:00" < current_time < "17:00") = 0.6;
TrustValue(current_time > "17:00" ) = 0.1...
```

Числовые значения уровней доверия, соответствующие определенному семантическому значению составляющей контекста, определяются экспертной оценкой, проведенной на основе анализа сервиса в интеллектуальном пространстве, для которого реализуется данный набор политик.

- 2) Второй тип правил используется при аутентификации и авторизации участника интеллектуального пространства. Правила представляют собой логические функции произведения соответствия уровней доверия компонент контекста, определенных первым типом правил, заданным интервалам, соответствующим какой-либо роли. Роль участника определяется множеством классов контекста  $o_{rl} \in O$  с допустимыми значениями атрибутов  $q_{rl_i}$ :  $(a_{rl_i}, b_{rl_i})$ ,  $0 < a_{rl_i} < b_{rl_i} < 1$ . Имеют вид логической функции:

$$f_{rl}(Context(t)) = \bigwedge_{i=0}^n (f_{trust_i}(o_{rl_i}, q_{rl_i}) \in (a_{rl_i}, b_{rl_i})) \quad (23)$$

Пример второго типа правил следующий:

```
Assign_role(some_rule) = (TrustValue(network) ∈ (0.8, 1)) &
(TrustValue(current_time) ∈ (0.3, 1)) & ...
```

- 3) Третий тип правил содержит права, закрепленные за каждой ролью. Эти правила применяются после присвоения участнику роли и определяют разрешенные действия для заданного типа ресурса. Использование ролей в брокере безопасности позволяет достаточно гибко настраивать наборы разрешенных действий для каждой категории участников. Правила, определяющие права доступа к ресурсу *res*, закрепленные за ролью *rl* задаются функцией:

$$f_{access}(rl, res) \rightarrow \{false, true\} \quad (24)$$

Пример третьего типа правил:

```
Permission(role, resource type);
```

Правила отображения атрибута класса объекта контекста в уровень доверия. Значения уровней доверия, соответствующие определенному значению атрибута класса объекта, определяются экспертной оценкой, проведенной на основе анализа проблемной области, для которой реализуются правила.

### 3.6 Выводы по главе 3

- 1) В главе представлена архитектура системы обеспечения инфомобильности, основанная на использовании технологии интеллектуальных пространств для реализации модели архитектуры «классная доска». Интеллектуальное пространство позволяет объединить разнородные источники информации и предоставить возможность совместной обработки этой информации.
- 2) Интеллектуальное пространство реализовано с помощью технологической платформы Smart-M3, разработанной по инициативе исследовательского центра компании Nokia для реализации платформы, способной соединить «информацию» из физического мира с возможностями обработки, представленными распределенными и повсеместными вычислениями. Аббревиатура M3 объединяет три ключевые идеи, лежащие в основе платформы: Multi-vendor, Multi-device, Multi-domain, то есть независимость от конкретных производителей, оборудования, области применения и возможность обмена информацией между различными программными модулями.
- 3) Для представления знаний в платформе Smart-M3 используется модель RDF, позволяющая задать описание связей понятий онтологий с помощью троек (субъект-предикат-объект). Подобное представление онтологий позволяет

представить онтологию в виде ориентированного графа, что может быть использовано для сопоставления разных онтологий. Выбор данных из хранилища осуществляется с помощью специализированного языка запросов SPARQL.

- 4) Сервис планирования мультимодальных маршрутов позволяет осуществлять планирование маршрутов как с использованием отдельных видов транспорта, так и комбинировать их в ходе следования по маршруту. При планировании маршрута по отдельным видам транспорта используется ориентированный граф, отражающий сеть дорог для соответствующего вида транспорта. В случае мультимодального маршрута сеть маршрутов отображается в мультиграф, в котором вершины соответствуют остановкам транспорта, а ребра – маршрутам между остановками, причем каждая пара остановок может быть соединена больше чем одним ребром. Планирование маршрута с совместным использованием личного транспорта осуществляется на основе имеющейся информации о существующих маршрутах водителей, являющихся пользователями системы и информации о планируемом маршруте потенциальных пассажиров. Для каждого потенциального пассажира подбирается водитель, удовлетворяющий следующим критериям: минимальное расстояние до точки встречи и от точки высадки до конечной точки маршрута для пассажира, минимальное время ожидания для пассажира и водителя, минимальное отклонение от кратчайшего пути для водителя.
- 5) Для обеспечения возможности свободного автоматического обмена знаний разработан сервис сопоставления онтологий. Его основная задача заключается в сравнении онтологий двух различных сервисов и выделении в них общих понятий для последующего обмена информацией с использованием разных онтологий. В случае невозможности автоматического сопоставления онтологий, сервис формирует задачи по сопоставлению онтологий для интернет-сообщества, используя технологию краудсорсинга.
- 6) Для ранжирования объектов, предоставленных сервисом поиска объектов, используется сервис формирования рекомендаций. Рейтинг объектов, на основе которого происходит ранжирование, вычисляется с помощью метода коллаборативной фильтрации, основанного на отношении между пользователями. В данном методе вычисляется вероятная оценка, которую пользователь мог бы поставить предлагаемому объекту. Вычисление проводится на основе оценок пользователей, имеющих схожие интересы, то есть одинаково оценивающих одни и те же объекты. При вычислении оценки дополнительно учитывается текущая ситуация.

- 7) Для обеспечения конфиденциальности информации о пользователях в систему обеспечения инфомобильности включен сервис, позволяющий разграничить доступ к соответствующей информации и обеспечивающий безопасный обмен при необходимости. Решение о предоставлении доступа вычисляется на основе контекста сервиса, запрашивающего доступ. Обмен информацией происходит посредством виртуального приватного интеллектуального пространства, которое является частью публичного ИП.

В следующей главе представлена реализация моделей и сервисов системы обеспечения инфомобильности, описанных в предыдущих главах.

## Глава 4. Реализация сервис-ориентированной системы обеспечения инфомобильности

### 4.1 Реализация сервиса планирования маршрутов

#### Планирование маршрутов по отдельным видам транспорта

В состав ГИС, используемой в системе обеспечения инфомобильности для планирования маршрутов, включена библиотека *pgRouting*, позволяющая прокладывать маршруты по карте, импортированной в базу данных под управлением *PostgreSQL*. Для корректной работы библиотеки требуется преобразование «сырых» данных путем построения топологии, позволяющей отразить дорожную сеть региона. Топология может быть представлена взвешенным ориентированным графом, к которому применимы основные алгоритмы поиска путей, реализованные в библиотеке.

Построение топологии осуществляется с использованием утилиты *osm2po*, позволяющей обработать исходный файл *OSM/XML* и преобразовать его в таблицу базы данных. В процессе преобразования из файла извлекается информация о выбранных классах дорог (пешеходные, автомобильные, магистрали и т.д.), определяются узловые точки (перекрестки, повороты), которые преобразуются в вершины графа и вычисляются расстояния между узловыми точками, определяющие вес соответствующих ребер. Кроме того, для каждого ребра заносится информация о средней разрешенной скорости движения и возможности реверсивного движения.

Каждая строка таблицы, в которой хранится топология и по которой производится поиск маршрута содержит ребро графа со следующими характеристиками:

- ID источника и назначения
- Класс дороги
- Длина пути в километрах
- Средняя разрешенная скорость движения по ребру
- Стоимость движения время, затрачиваемое на перемещение по ребру
- Стоимость движения в обратном направлении (время движения в обратном направлении). Позволяет задавать возможность движения по ребру в обе стороны.

При планировании маршрута с использованием одного типа транспорта используется алгоритм A\*. Выбор алгоритма обоснован использованием эвристики, позволяющей сократить область выбора вершин за счет дополнительной оценки кратчайшего расстояния и ускорить тем самым поиск кратчайшего пути. Критерием при построении пути является стоимость движения

по ребру. *SQL*-запрос, позволяющий получить кратчайший путь с помощью алгоритма A\*, реализованного в библиотеке pgRouting выглядит следующим образом:

```
SELECT * FROM pgr_astar('SELECT id,
                        source::integer,
                        target::integer,
                        cost,
                        reverse_cost,
                        x1, y1, x2, y2
                        FROM ways'
, origin, destination, true , true);
```

Листинг 3 — Вызов функции A\* для планирования маршрута

Аргументами данного запроса являются подзапрос, позволяющий выбрать границы, в которых будет производиться построение маршрута, начальная и конечная точки, указание, является ли граф ориентированным и учет возможности движения в обратном направлении. Представленный в листинге 3 запрос при поиске маршрута выбирает всю преобразованную область карты и учитывает все типы дорог.

При поиске маршрута по отдельным видам транспорта, используемым в системе обеспечения инфомобильности используются фильтры, позволяющие выбрать для построения маршрута только те дороги, по которым может двигаться выбранный тип транспорта. Для этого ко внутреннему запросу добавляется условие, позволяющее ограничить используемые классы дорог. При импорте в файле конфигурации *osm2po* было установлено соответствие классов дорог, представленных в файле *OSM*, индексов, назначенных при импорте и типов используемых транспортных средств [150], отображенное в таблице 4.

Согласно таблице 4, при поиске пути для проезда на автомобиле следует выбирать дороги, ограниченные классами от 11 до 43, а для пешего маршрута — от 31 до 92. При этом время, затрачиваемое на поиск маршрута, сокращается примерно в 1,3 раза для пешего маршрута и в 4,8 раза для автомобильного маршрута (см. рисунок 35). Данная разница обусловлена тем, что количество ребер, принадлежащим классам, пригодным для пешеходов значительно превышает, количество ребер, принадлежащих классам, пригодным для автомобиля. Для карты Санкт-Петербурга, используемой для проведения тестов, в графе дорог выделяется 317012 ребер, доступных для пешехода, против 119828 ребер, доступных для автомобиля. Всего в графе было выделено 406666 ребер.

Таблица 4 — Типы дорог, учитываемые при планировании маршрута

Класс в файле OSM	Класс в таблице БД	Виды транспорта	Комментарий
motorway motorway_link	11 12	Автомобиль	Автомагистраль и съезды с нее
trunk trunk_link	13 14	Автомобиль	Междугородные маршруты, не прерывающиеся в городе
primary primary_link	15 16	Автомобиль	Дороги регионального значения, центральные магистрали городов, соответствующие развязки
secondary secondary_link	21 22	Автомобиль	Дороги областного значения, основные магистрали районов города, соответствующие развязки
tertiary tertiary_link	31 32	Автомобиль Велосипед	Другие важные дороги, микрорайонные транзитные улицы, соответствующие развязки
residential	41	Автомобиль Велосипед	Дороги внутри жилых зон
road	42	Автомобиль Велосипед	Возможные дороги
unclassified	43	Автомобиль Велосипед	Остальные дороги местного значения
service	51	Автомобиль Велосипед	Служебные проезды
pedestrian	62	Пешеход Велосипед	Улицы и площади
living_street	63	Автомобиль, Велосипед, Пешеход	То же, что и residential, за исключением приоритета пешеходов перед другими ТС
track	71	Велосипед Пешеход	Лесные дороги, неофициальные грунтовые дороги
path	72	Велосипед Пешеход	Тропы
cycleway	81	Велосипед	Велодорожки
footway	91	Пешеход	Пешеходные дорожки, тротуары
steps	92	Пешеход	Лестницы, лестничные пролеты

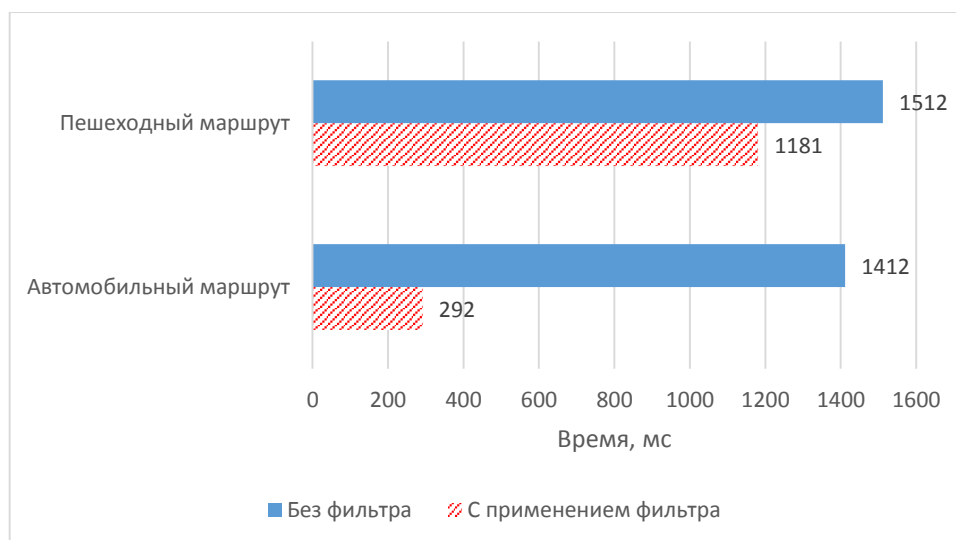


Рисунок 35 — Сокращение времени поиска при применении фильтров по классам дорог

Результатом поиска маршрута является последовательность вершин, по которым следует совершать движение и общая протяженность пути. Данная информация может быть сразу передана пользователю через интеллектуальное пространство, либо использоваться в сервисе планирования мультимодальных маршрутов, реализация которого описана в следующем разделе.

### Планирование мультимодального маршрута

Как было отмечено в 3.3, при построении мультимодального маршрута приоритет отдается использованию общественного транспорта и пешим маршрутам. Данные о маршрутах и расписаниях общественного транспорта распространяются в общедоступном формате *GTFS*. Данный формат содержит файлы, повторяющие требуемую структуру таблиц базы данных. Полное описание формата представлено в [80]. Благодаря своей структуре данные в формате *GTFS* могут быть импортированы в базу данных для обеспечения возможности быстрого поиска маршрутов и выборки различных данных о маршруте.

Для Санкт-Петербурга данные о маршрутах общественного транспорта в формате *GTFS* предоставляются порталом общественного транспорта [17]. Дополнительно к статичному расписанию данный портал предоставляет текущее положение транспортных средств, которое может быть использовано для прогнозирования времени прибытия ближайшего транспортного средства на выбранную остановку.

После импорта данных о маршрутах общественного транспорта по г. Санкт-Петербург в базу данных под управлением *PostgreSQL* и формирования мультиграфа, был проведен их анализ. Всего по городу насчитывается 6962 остановки, между которыми проходит 965 маршрутов. Количество ребер мультиграфа, используемых для отображения маршрутов ОТ, составляет 19773. На рисунке 36 изображены маршруты (а) и остановки (б), полученные при обработке исходных файлов расписания движения ОТ для Санкт-Петербурга.



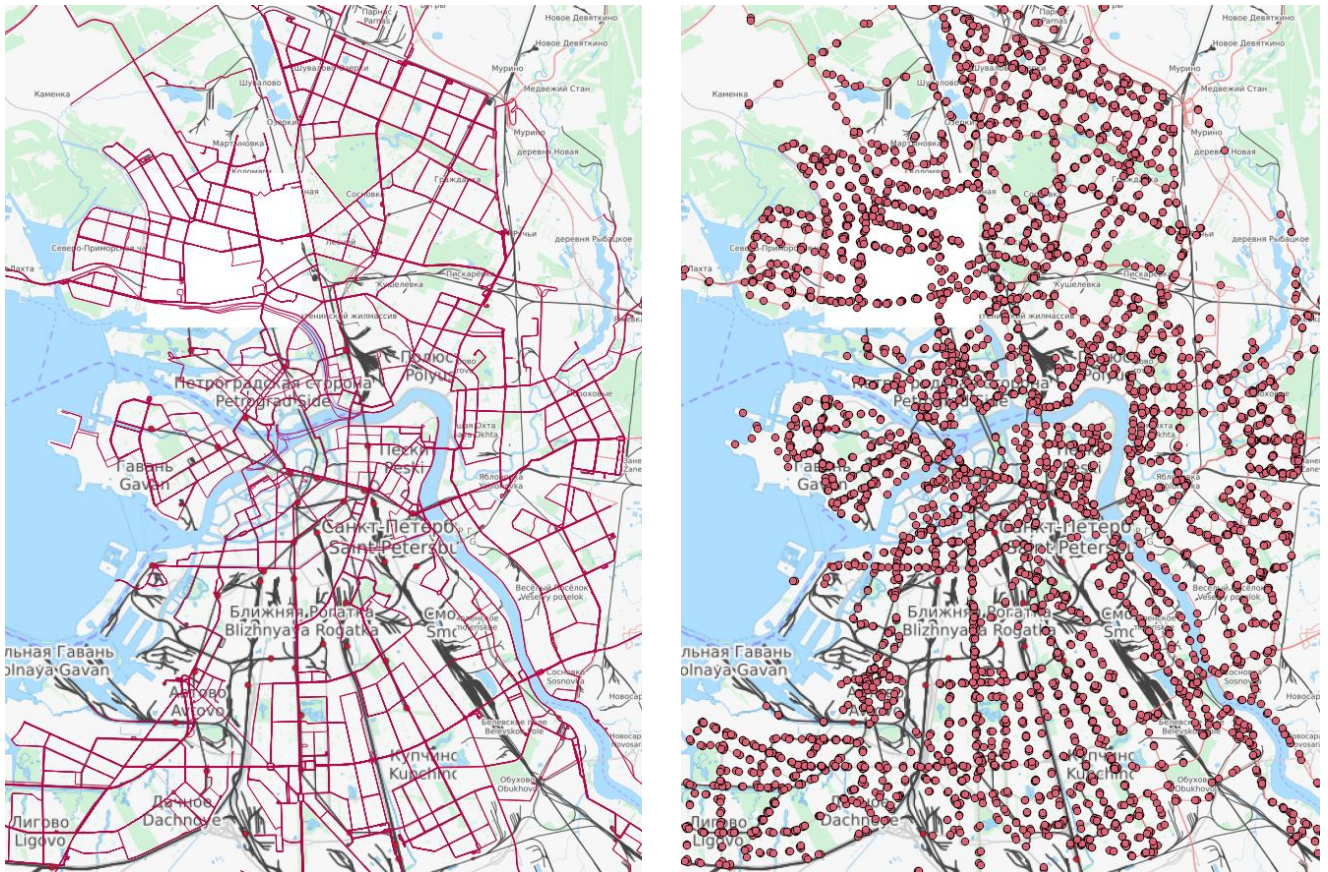


Рисунок 36 — а) маршруты ОТ, б) остановки ОТ

При выборе точек начала и конца маршрута пользователь указывает их на карте без привязки ее к полученному графу. Таким образом, первой задачей, необходимой при планировании мультимодального пути является создание фиктивных вершин и ребер, соединяющих новые вершины с ближайшими вершинами уже существующего графа. Для этого применяется *SQL*-запрос, представленный в листинге 4:

```
WITH closest_candidates AS (
  SELECT
    s1.stop_id,
    CAST (st_distance_sphere(s1.geom, s2.geom) AS INT) AS distance,
    to_char(time '12:00:00' + (st_distance_sphere(s1.geom,
s2.geom)::int/1.389 || ' seconds'))::interval, 'HH24:MI:SS')::varchar(255)
AS walk_time FROM
  stops as s1
  LEFT JOIN stops as s2 on CAST (s2.stop_id AS INTEGER) = 18446
  WHERE CAST (st_distance_sphere(s1.geom, s2.geom) AS INT) < 300
  ORDER BY s1.geom <-> s2.geom
  LIMIT 100);
SELECT stop_id, distance
FROM closest_candidates
WHERE distance < 200
ORDER BY distance;
```

Листинг 4 — Запрос для создания фиктивной вершины и ребер

С помощью данного запроса из БД, в которой хранится мультиграф, извлекаются ближайшие вершины в радиусе, определенном расстоянием, которое пользователь согласен пройти пешком (*Detour*) и рассчитываются расстояния и время движения до них, которые используются в качестве весов новых ребер.

На каждой итерации алгоритма Дейкстры, реализованного согласно описанию в листинге 1, после выбора вершины с минимальным временем движения из списка доступных вершин в графе производится поиск соседних вершин, достижимых с использованием общественного транспорта, и вершин, достижимых пешком.

```
WITH closest_candidates AS (
  SELECT
    s1.stop_id,
    CAST (st_distance_sphere(s1.geom, s2.geom) AS INT) AS distance,
    to_char(time '12:00:00' + (st_distance_sphere(s1.geom,
s2.geom)::int/1.389 || ' seconds')::interval, 'HH24:MI:SS')::varchar(255)
AS walk_time
  FROM stops as s1
  LEFT JOIN stops as s2 on CAST (s2.stop_id AS INTEGER) = 18446
  WHERE CAST (st_distance_sphere(s1.geom, s2.geom) AS INT) < 300
  ORDER BY s1.geom <-> s2.geom
  LIMIT 100)

SELECT s1.trip_id as trip_id, s1.departure_time as departure_time,
s2.stop_sequence, s2.stop_id as stop_id, s2.arrival_time
FROM stop_times as s1
  LEFT JOIN stop_times as s2 ON s1.trip_id = s2.trip_id and
s1.stop_sequence +1 = s2.stop_sequence
WHERE CAST (s1.stop_id AS INTEGER) = 18446 AND s1.departure_time IS NOT
NULL AND s1.departure_time >= '12:00:00' AND s1.departure_time <=
'12:15:00'

UNION

SELECT '-1' as trip_id, '12:00:00' as departure_time, '0', cl.stop_id as
stop_id, cl.walk_time--, cl.distance
FROM closest_candidates as cl
WHERE cl.distance > 0
ORDER BY departure_time ASC
```

Листинг 5 — Выбор вершин, достижимых из текущей вершины

Для вершин, достижимых пешком, также создаются фиктивные ребра, в качестве весов имеющие расстояние между вершинами и время движения между ними. Данное решение позволяет рассматривать новые ребра и включать в общий маршрут пешеходные участки без изменения алгоритма. При формировании маршрута в путь до каждой вершины заносится ребро с минимальным весом и вершина, являющаяся источником ребра.

При достижении конечной вершины общий путь формируется как последовательность вершин и ребер, которые пользователь должен использовать в ходе следования маршрута. Вершины, в которых изменяется номер маршрута, или которые являются источниками для пешеходного маршрута отмечаются как вершины пересадки.

Измерение времени работы алгоритма производилось путем поиска фиксированного маршрута с постепенным увеличением размера рассматриваемой алгоритмом области: от Васильевского Острова, на котором расположено 1656 остановок, до всего Санкт-Петербурга, в котором, как было упомянуто ранее, насчитывается 6962 остановки. На каждом из этапов проводилось 100 измерений среди которых было вычислено среднее значение и в качестве диапазона погрешности взята процентиль 0.9.

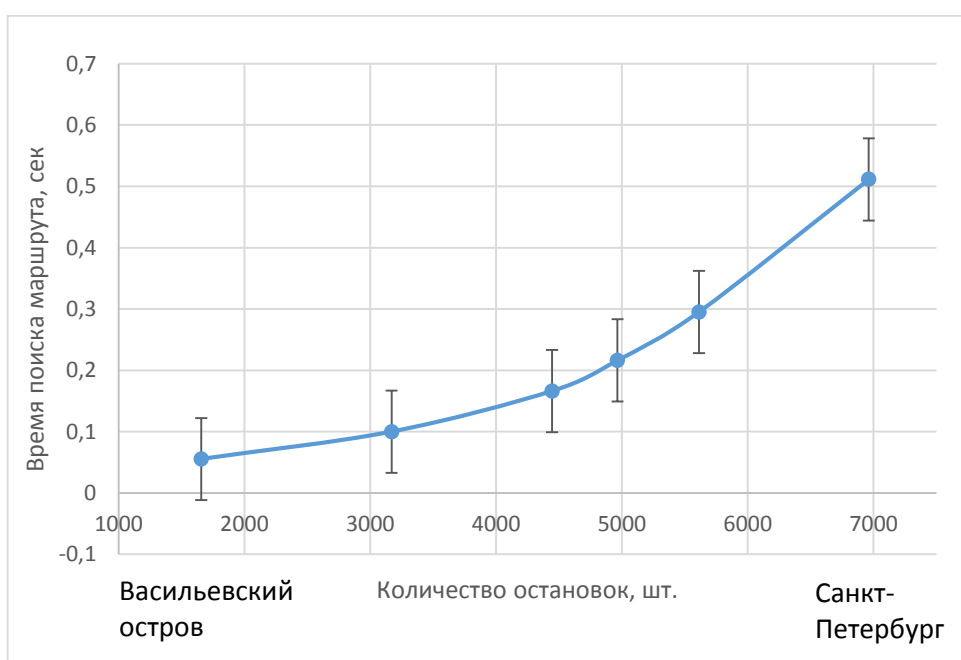


Рисунок 37 — Время поиска мультимодального маршрута

### Планирование поездок с совместным использованием личного автотранспорта

При реализации возможности планирования совместной поездки с использованием личного транспорта использовались наработки, описанные в предыдущих разделах текущего параграфа: поиск точек, ближайших к указанному местоположению пользователя и алгоритм поиска маршрута с использованием фильтра типов дорог.

Планирование поездки с использованием личного транспорта так же имеет свои особенности реализации, связанные с необходимостью оптимизации расчета большого количества комбинаций точек встречи и высадки пассажиров [159].

Первой особенностью является поиск потенциальных точек встречи на графе дорог. Запрос, представленный в листинге 4, позволяет найти ближайшие точки в определенном радиусе, без учета их достижимости. Другими словами, расстоянием между точками вычисляется

на сфероиде, отражающем форму Земли, при этом топология дорог не учитывается, и в окружность, ограниченную выбранным радиусом могут попасть точки, пусть до которых окажется длиннее, чем ограничение, заданное пользователем. Проверка каждой найденной точки алгоритмом Дейкстры или  $A^*$  требует серьезных временных затрат, в связи с чем, для выбора точек используется функция библиотеки *pgRouting* — *pgr\_drivingDistance* — позволяющая построить полигон, заключающий в себя точки, путь до которых не превышает указанного значения. На рисунке 38 представлен пример ситуации, в которой использование функции *pgr\_drivingDistance* позволяет исключить точки, попадающие в окружность радиуса *Distance* с центром в текущем местоположении пользователя (точке А), но действительный путь до которых из этой точки больше чем параметр *Distance* (точки G, F, I). В результате работы функции остаются только четыре потенциальные точки встречи (A, D, L, H).

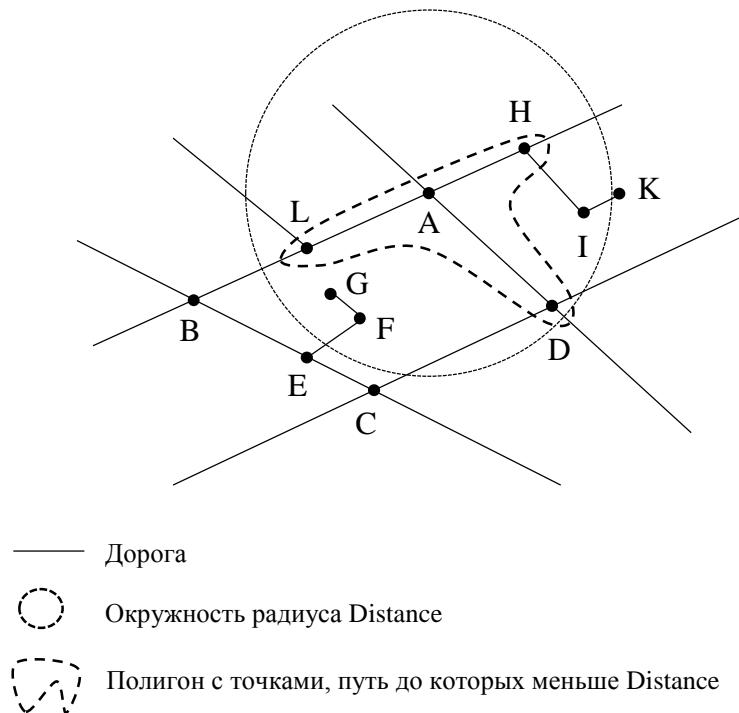


Рисунок 38 — Результат работы функции *pgr\_drivingDistance*

Второй является расчет маршрутов между одной начальной и несколькими конечными точками. На этапе перебора вариантов совместной поездки требуется рассчитать множество путей водителя до потенциальных точек встречи, множество комбинаций путей от потенциальных точек встречи до точек высадки и множество путей от точек высадки до следующей точки пути водителя. Применение на каждом этапе алгоритма Дейкстры или  $A^*$  требует значительных вычислительных и временных затрат, в связи с чем требуется оптимизация расчета комбинаций путей на каждом из этапов. Возможность такой оптимизации предоставлена в библиотеке *pgRouting* и реализована в функции *pgr\_kdijkstraCost*. Данная функция позволяет найти длину пути от выбранной точки до массива точек за один проход алгоритма Дейкстры. При

этом чем больше точек указано в массиве, тем больший выигрыш получается от использования данной функции. Сравнение времени выполнения для восьми точек назначения с применением фильтров и без них приведено на рисунке 39.

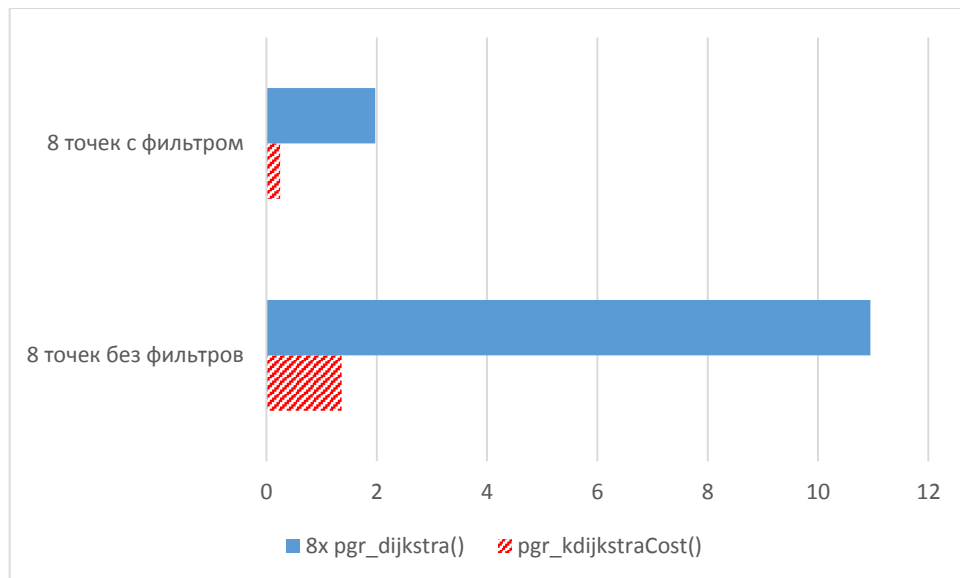


Рисунок 39 — Сравнение времени расчета расстояния для восьми точек с использованием алгоритма Дейкстры и с использованием алгоритма *kDijkstra*

Третьей особенностью является необходимость кеширования маршрутов. При наличии большого количества пользователей, планирующих совместную поездку часть маршрутов, построенных при вычислении совпадающего пути может пересекаться. Для снижения нагрузки на сервер применяется кеширование недавно рассчитанных маршрутов.

Одна из особенностей планирования совместной поездки с использованием личного транспорта заключается в контроле свободных мест в используемом транспорте и назначении водителей пассажирам. После выбора пассажиром пути, наиболее удовлетворяющего его критериям, осуществляется изменение характеристик точек, находящихся между выбранными точками встречи и высадки, происходящее следующим образом:

Для пассажира в значение параметра *driveByVehicle* записывается ID водителя, который согласен его подвезти:

```
('user2point*', 'drivebyvehicle', 'FALSE') → ('user2point*', 'drivebyvehicle', 'user1')
```

Число свободных мест в транспорте, закрепленном за водителем уменьшается на количество пассажиров, осуществляющих поездку по выбранному отрезку пути:

```
('user1point1', 'vacantseats', '3') → ('user1point1', 'vacantseats', '2')
```

В связи с тем, что в путь водителя добавляется новая точка, необходимо изменение параметра *previousPoint* соседних точек относительно новой. В параметр *previousPoint*

добавленной точки копируется идентификатор из параметра *previousPoint* следующей за ней точки, а в параметр *previousPoint* следующей точки записывается идентификатор добавленной точки, т.е.:

```
p[1], ... , p[i], p[i+1], ... , p[n] // начальный список точек. i – индекс
точки, после которой будет добавлена новая точка.
p[1], ... , p[i], p[i+1], p[i+2], ... p[n+1] // список после добавления,
p[i+1] – новая точка.
p[i+1].previousPoint = p[i]
p[i+2].previousPoint = p[i+1]
```

Все изменения вносятся в интеллектуальное пространство и используются при последующих расчетах путей пользователей. Пользователи, запросившие планирование совместной поездки, оповещаются о результате и могут начать совместную поездку, в случае если результат положительный.

Тестирование быстродействия при планировании совместного использования личного транспорта проводилось на тестовом стенде, имеющем следующую конфигурацию: центральный процессор Intel Core i7 с тактовой частотой 3,4 ГГц, и оперативной памятью — DDR3 8 Гб. Испытания представляли собой прогон алгоритма на случайных значениях координат точек начала и конца, выбранных на карте Санкт-Петербурга, для заданного числа водителей и пассажиров. Результаты тестовых испытаний представлены в таблице и на рисунке 40.

Таблица 5 — Результаты тестовых испытаний

<b>Число водителей</b>	<b>Число пассажиров</b>	<b>Время планирования совпадающего пути</b>
1	1	0,0135
5	5	0,0316
10	10	0,0641
20	20	0,2248
40	40	1,5462
60	60	2,2416
80	80	3,4725

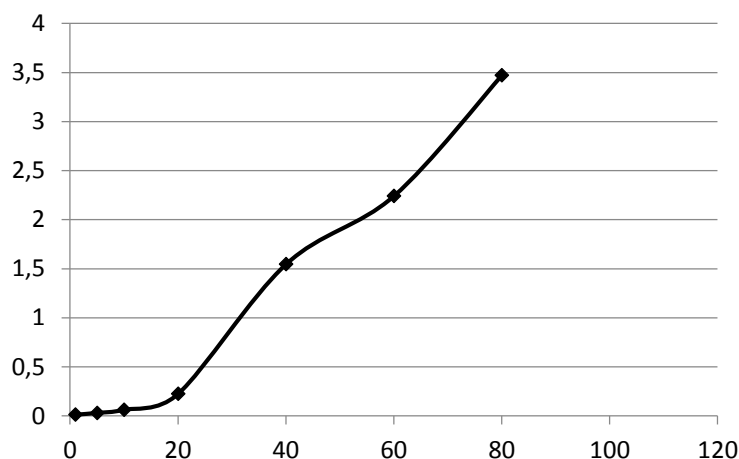


Рисунок 40 — Зависимость времени выполнения от количества пользователей

Быстродействие было достигнуто в основном за счет использования кеширования и расчета путей между массивом точек. Важную роль также имеет использование эвристик, позволившее уменьшить количество потенциальных точек встречи и высадки, что уменьшило количество их комбинаций.

#### 4.2 Реализация ГИС для сервисов планирования маршрута и поиска объектов

В соответствии с анализом существующих ГИС, представленным в параграфе 1.4, ГИС, основанная на данных проекта *OpenStreetMap*, наиболее полно удовлетворяет требованиям, выдвигаемым системой обеспечения информативности к ГИС. Использование карт *Google* является затруднительным вследствие того, что *Google* имеет серьезное дневное ограничение на число прокладываемых маршрутов, а также для некоторых стран, в число которых входит Россия, детализованные карты доступны только для крупных городов.

Для снижения нагрузки на главный сайт *OpenStreetMap*, разработчикам рекомендуется использовать зеркала, либо организовывать собственный сервер для визуализации и распространения карт конечным пользователям. Предоставление карты на сайте *OpenStreetMap* организовано следующим образом. Для хранения данных используется СУБД *PostgreSQL* [135] с расширением *PostGIS* [136], предоставляющим функции для работы с геоданными. Преобразование данных карт из базы данных в графический формат реализовано с помощью программы *Mapnik* [112]. В качестве веб-сервера для распространения карт используется веб-сервер *Apache* с модулем *mod\_tile*. Данный модуль позволяет настроить политику кеширования участков карты и обеспечить их распространение конечным пользователям. Для отображения карты на веб-странице проекта используется *JavaScript* библиотека *Leaflet* [107]. Все программное обеспечение, представленное выше имеет открытый исходный код и может быть использовано всеми для организации картографических сервисов.

Использование серверов проекта, как и в случае с картами *Google* накладывает ряд ограничений на разработчиков, поэтому для системы обеспечения инфомобильности используется собственный сервер. Обоснование выбора используемых библиотек и описание организации их совместной работы представлено далее.

### **Организация хранилища исходных данных для карт**

Распространение картографических данных проекта *OpenStreetMap* осуществляется в виде *XML*-файлов. Структура файла карты *OpenStreetMap XML* представлена в листинге 6. Листинг содержит следующие главные элементы *XML* структуры карты: узлы (*node*), пути (*way*) и отношения (*relation*). Узел состоит из одиночной геопространственной точки, определяемой широтой и долготой, и может быть использован для определения свойств точки с помощью тегов. Путь представляется упорядоченным списком узлов, которые имеют, как минимум, один тег или включены в состав отношения. В пути может быть от 2 до 2000 узлов. Отношение состоит из одного и более тегов, упорядоченного списка одного или более узлов и/или путей, как членов, которые используются для определения логического или географического отношения между другими элементами. Файл также содержит описание свойств всех элементов, которые могут быть отображены на карте с дополнительной информацией о том, когда и кем элемент был добавлен и т.д.

Представленная структура файла позволяет свободно манипулировать данными путем конфигурации стиля карты и создания слоев, на которых будет отображена дополнительная информация поверх основного слоя карты. Однако использование прямого доступа к файлу при решении задачи визуализации карты существенно снижает скорость доступа к данным и визуализации карты из-за того, что файл имеет достаточно большой размер. Например, только карта Санкт-Петербурга занимает около 350 Мб дискового пространства. Для хранения данных обо всей планете требуется уже более 450 Гб свободного дискового пространства (в двоичном формате файл занимает около 30 Гб, что также является неприемлемым для организации доступа к данным).

Для решения проблемы, связанной с хранением большого файла и быстрого доступа к данным в нем, сообщество *OSM* была разработана модель хранения данных в СУБД *PostgreSQL*. Схема *osm2pgsql* исторически была стандартным способом импорта данных *OSM* в СУБД *PostgreSQL*, имеющую расширение *PostGIS* для работы с геоинформацией, для последующего использования [131]. Импорт осуществляется с использованием одноименной утилиты, которая поддерживает два режима работы, определяющие использование оперативной памяти в процессе импорта: сокращенный и несокращенный. Сокращенный режим позволяет использовать меньшее количество оперативной памяти, поддерживает обновление карты, но



время, требуемое для импорта карты, существенно зависит от скорости работы жесткого диска и даже на достаточно производительном оборудовании может занять до нескольких дней, если требуется импорт карты всего мира. Несокращенный режим работает быстрее, но не предоставляет поддержку обновлений и требует огромного количества памяти.

```
<bounds minlat="54.0889580" minlon="12.2487570"          maxlat="54.0913900"
maxlon="12.2524800"/>
<node id="1" version="1" changeset="1" lat="54.0900666"
  lon="12.2539381" user="user1" uid="10" visible="true"  timestamp="">
  <tag k="name" v="Neu Broderstorf"/>
  <tag k="traffic_sign" v="city_limit"/>
</node>
...
<way id="110" user="user2" uid="11" visible="true" version="5"
changeset="4" timestamp="">
  <nd ref="292403538"/>
  <nd ref="298884289"/>
  ...
  <nd ref="261728686"/>
  <tag k="highway" v="unclassified"/>
  <tag k="name" v="Pastower Straße"/>
</way>
<relation id="5" user="user3" uid="12" visible="true"      version="28"
changeset="6" timestamp="">
  <member type="node" ref="294942404" role=""/>
  ...
  <member type="node" ref="364933006" role=""/>
  <member type="way" ref="4579143" role=""/>
  ...
  <member type="node" ref="249673494" role=""/>
  <tag k="name" v="Küstenbus Linie 123"/>
  <tag k="network" v="VWV"/>
  <tag k="operator" v="Regionalverkehr Küste"/>
  <tag k="ref" v="123"/>
  <tag k="route" v="bus"/>
  <tag k="type" v="route"/>
</relation>
```

Листинг 6— Структура файла с картографическими данными

Процесс импорта происходит с потерями и контролируется *XML*-файлом конфигурации, в котором задаются элементы, которые должны быть импортированы. Значения этих элементов импортируются как колонки в таблицах точек, линий и полигонов. Значения всех тегов могут быть импортированы в специальную колонку, имеющую тип “*hstore*”. В связи с тем, что таблицы могут быть достаточно большими, требуется индексация содержимого.

## Визуализация карты

Для визуализации карты используется набор утилит *Mapnik* [112]. Он может брать данные для визуализации карты из множества источников: напрямую из файла *OSM-XML*, из базы данных, из файлов формы *ESRI* и других. Как было упомянуто в начале раздела, данный набор утилит используется на главной странице проекта *OpenStreetMap*. *Mapnik* написан на языке *C++* и может быть использован в программах, разработанных на других языках программирования, таких как *JavaScript (Node.js)*, *Python*, *Ruby*, *Java*. Он использует библиотеку визуализации «*Anti-Grain Geometry*» и предоставляет сглаженную визуализацию с субпиксельной точностью. Установка *Mapnik* возможна как на всех системах на базе ядра *Linux*, так и в операционных системах *OS X* и *Microsoft Windows*, для которых созданы установочные готовые пакеты.

*Mapnik* осуществляет вывод изображений карты во многих графических форматах — *PNG*, *JPEG*, *SVG*, и *PDF* — в любом заданном размере, любой области в любой географической проекции. Чаще всего используется визуализация маленьких плиток — тайлов, размером  $256 \times 256$  пикселей, которые затем объединяются в «скользящую карту» с использованием *JavaScript*-библиотек.

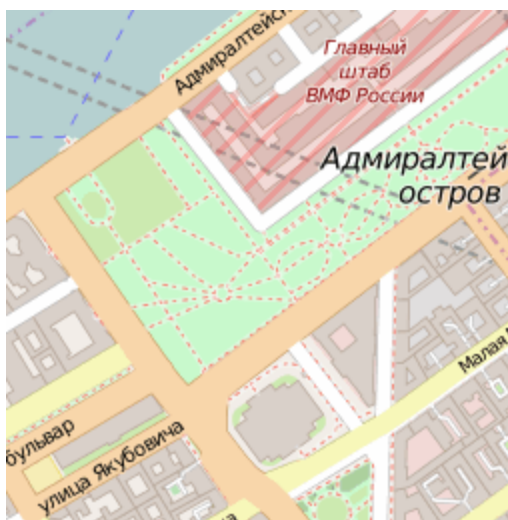


Рисунок — Пример плитки для Санкт-Петербурга на уровне приближения 15

Для организации сервера, распределяющего плитки между пользователями используется модуль *mod-tile* для веб-сервера *Apache*. Этот модуль использует *Mapnik* для генерации плиток «на лету». Он предоставляет динамическую комбинацию эффективного кэша и генерации новых плиток. Благодаря динамической визуализации на диске необходимо держать небольшое количество готовых плиток, что позволяет сэкономить место. В то же время, стратегия хранения кэша позволяет предоставить высокую производительность раздачи и может поддерживать несколько тысяч запросов в секунду.

Для объединения отдельных плиток в целую «скользящую» карту [155] существует две основных библиотеки, написанных на языке *JavaScript: OpenLayers* и *Leaflet* [107]. Карта может быть встроена в веб-страницу с элементами управления, такими как уровень приближения (зума), текущий масштаб и кнопки перемещения по карте. Дополнительно библиотеки предоставляют программные интерфейсы для отображения слоев с дополнительной информацией поверх основного слоя карты.

1) *OpenLayers*. Данная библиотека была разработана сообществом *OpenStreetMap* для отображения карты на главной странице проекта. Функции, предоставляемые библиотекой, позволяют тонко настроить все свойства карты. С ее помощью можно отображать карты, полученные из любых источников, и дополнительные объекты, такие как маркеры, линии, многоугольники, всплывающие окна и т.д. Третье поколение библиотеки (версии 3.x) было существенно переработано для повышения быстродействия на мобильных устройствах и предоставления более простого и понятного *API*. Библиотека содержит функции для управления картой, анимации перемещения, масштабирования и многое другое.

2) *Leaflet*. Данная библиотека была разработана сообществом проекта *CloudMade*. Она реализует большинство функций, предоставляемых *OpenLayers*, предоставляя более простой *API*. Кроме того, за счет отказа от части редко используемых функций, библиотека *Leaflet* занимает меньше дискового пространства и обеспечивает высокую скорость работы. Она одинаково быстро работает как на стационарных компьютерах, так и на мобильных устройствах, предоставляя поддержку как современных технологий *HTML5* и *CSS3* на современных браузерах, так и обеспечивая обратную совместимость со старыми. За счет поддержки широкого спектра устройств и быстродействия, библиотека *Leaflet* в настоящее время используется для отображения карты на главной странице проекта *OpenStreetMap*, а также во многих других проектах, например, *Flickr*, *Wikipedia mobile apps*, *Foursquare*, *craigslist*, *IGN*, *Washington Post*, *The Wall Street Journal*, *Geocaching.com*, *City-Data.com*, *StreetEasy*, *Nestoria*, *Skobbler* и многих других.

### **Дополнительные функции, предоставляемые сторонними библиотеками**

К настоящему времени сообществом *OpenStreetMap* было разработано множество библиотек, позволяющих обрабатывать данные карт *OSM*. Наиболее часто используемые функции любой ГИС – поиск маршрутов и геокодирование.

Данные проекта *OpenStreetMap* включают в себя информацию о дорогах, что позволяет обеспечить поиск маршрута с использованием различных транспортных средств: пешеходные, автомобильные, велосипедные и даже конные маршруты. Для обеспечения возможности поиска маршрута данные должны быть преобразованы в граф. К настоящему времени было разработано

множество библиотек для преобразования данных, которые также предоставляют функции для навигации по ним. Среди этого множества наиболее интересными и функциональными являются библиотеки *pgRouting*, *GraphHopper* и *osm2po*.

1) *pgRouting* [133] является расширением для СУБД *PostgreSQL*. Она позволяет обеспечить работу с данными, импортированными в базу данных под управлением *PostgreSQL* с расширением *PostGIS*. Перед использованием функций поиска маршрутов библиотека *pgRouting* производит построение топологии по импортированной карте и создает граф связей между узловыми точками, по которым совершается последующее планирование маршрута. Основной код библиотеки написан на языке C++ с обертками для использования *SQL*.

*pgRouting* предоставляет реализацию следующих алгоритмов:

- Алгоритм Джонсона для поиска кратчайших путей между всеми парами вершин;
- Алгоритм Флойда-Уоршелла для поиска кратчайших путей между всеми парами вершин;
- Алгоритм Дейкстры для поиска кратчайшего пути;
- Алгоритм поиска кратчайшего пути A\*;
- Двухнаправленный алгоритм Дейкстры для поиска кратчайшего пути;
- Двухнаправленный алгоритм A\* для поиска кратчайшего пути;
- Алгоритм Driving Distance для поиска вершин, расположенных на определенном расстоянии от начальной вершины. Расстояние рассчитывается как путь, который преодолевается до вершины.
- Алгоритм K-Shortest path для поиска нескольких путей между заданными вершинами;
- Алгоритм K-Dijkstra для поиска путей между одной начальной и несколькими конечными точками;
- Алгоритм для решения задачи коммивояжера;
- *Turn Restriction Shortest Path (TRSP)* – алгоритм для поиска кратчайшего пути с учетом запретов на маневры;

2) *GraphHopper* [86] является быстрым и эффективным —с точки зрения затрат памяти — инструментом для поиска маршрута. Разработчику предоставляется библиотека, написанная на Java, с помощью которой можно организовать собственный сервер для поиска маршрутов. Версия библиотеки для Android позволяет встраивать ее в приложение, но требует затрат памяти на хранение карты и не отличается высоким быстродействием в силу ограниченной вычислительной

мощности мобильных устройств. По умолчанию работает с данными *OpenStreetMap* в форматах \*.osm и \*.pbf, но может быть адаптирован к использованию других данных.

*GraphHopper* предоставляет реализации следующих алгоритмов:

- Алгоритм Дейкстры для поиска кратчайшего пути;
- Алгоритм поиска кратчайшего пути A\*;
- Двухнаправленный алгоритм Дейкстры для поиска кратчайшего пути;
- Двухнаправленный алгоритм A\* для поиска кратчайшего пути;
- Алгоритм для поиска путей между одной начальной и несколькими конечными точками;

3) *osm2po*. Включает в себя конвертер и инструменты для поиска маршрута. Позволяет быстро обработать большие файлы в формате \*.osm. Конвертированные данные могут быть импортированы в базу данных под управлением *PostgreSQL*, причем импорт осуществляется в виде, пригодном для навигации с использованием функций пакета *pgRouting*. *Osm2po* также имеет свое ядро для поиска маршрута, которое включает в себя алгоритм Дейкстры.

Для геокодирования также разработано несколько решений, среди которых наиболее значимыми являются следующие:

1) *Tiger Geocoder*. Геокодер, основанный на *PL/pgSQL*, создан для работы с системой *TIGER (Topologically Integrated Geographic Encoding and Referencing system)*. Данная система используется статистическим бюро США. Геокодер включен в пакет расширения *PostGIS* и может использоваться с БД под управлением *PostgreSQL*. В настоящее время может использоваться только на территории США, хотя есть исследования, позволяющие адаптировать схему данных *TIGER* к другим странам.

2) *Nominatim*. Геокодер для *PostGIS*, набирающий популярность и пригодный для международного использования. Поставляется в виде веб-сервиса, который разработчики могут организовать на своем оборудовании. Позволяет осуществлять прямой и обратное геокодирование. Требуется предварительную обработку данных.

### **Организация картографического сервиса для системы обеспечения инфомобильности**

Для удовлетворения данных требований при организации ГИС была выбрана следующая комбинация проектов из числа представленных выше:

- *OpenStreetMap* как источник картографических данных;
- СУБД *PostgreSQL* с расширением *PostGIS* были выбраны для хранения карты,
- Модуль *mod\_tile* веб-сервера *Apache* с визуализатором карт на основе *Mapnik* для визуализации карт и распространения между пользователями,

- Библиотека *pgRouting* для планирования маршрутов,
- Библиотека *Nominatim* для геокодирования.

Взаимодействие всех элементов картографического сервиса показано на рисунке 41. Для отображения карты на стороне клиента используется библиотека *Leaflet*. Текущее местоположение пользователя предоставляется с использованием специфичного для устройства *API* для геолокации.

Для импорта картографической информации из XML файла в базу данных *PostgreSQL* используются две утилиты *osm2pgsql* и *osm2po*. Первая утилита импортирует весь файл OSM XML в таблицы базы данных, за исключением информации об изменениях данных, версиях, временных штампах и т.д. Она в автоматическом режиме создает требуемые таблицы, заполняет их информацией и создает индексы для повышения скорости работы с информацией.

Утилита *osm2po* используется для преобразования данных OSM в граф, пригодный для навигации и его последующего импорта в базу данных под управлением *PostgreSQL*. Данная утилита автоматически создает все требуемые таблицы, заполняет их данными и создает топологию по типам дорог, указанных в конфигурационном файле.

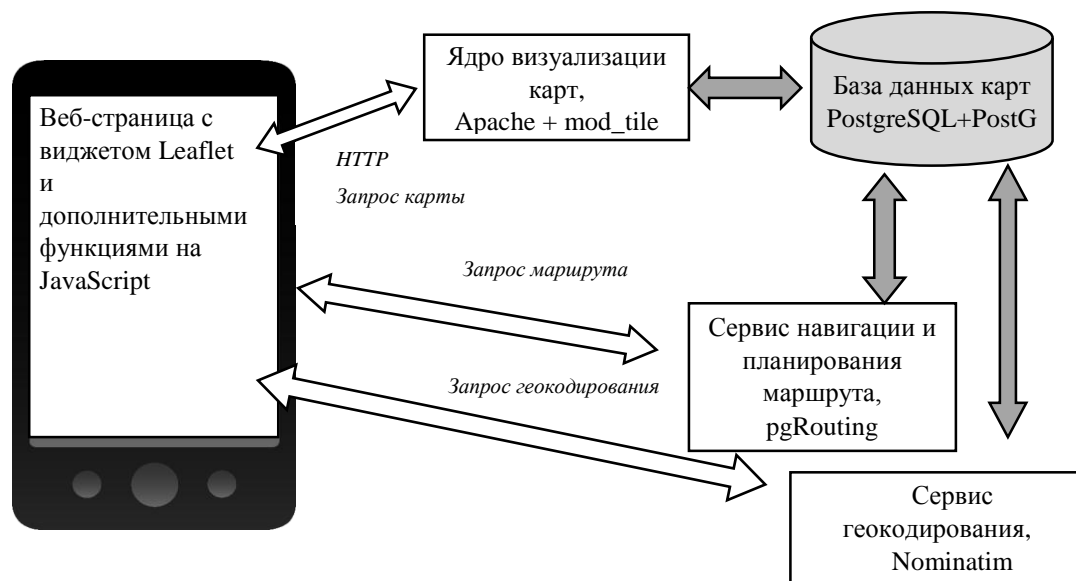


Рисунок 41 — Архитектура картографического сервиса для системы обеспечения инфомобильности

Взаимодействие с клиентом осуществляется с помощью веб-страницы в которую с помощью библиотеки *Leaflet* встроена «скользящая» карта. Карта инициализируется при открытии страницы пользователем и состоит из множества плиток, содержащих визуализированные данные карты с определенным уровнем приближения (уровнем зума). Плитки генерируются на сервере демоном *renderd*, который является частью модуля *mod\_tile*. Генерация плитки является достаточно затратной процедурой, в связи с чем, для оптимизации

нагрузки на сервер уже готовые плитки сохраняются в кэш, структура которого показана в листинге

```
shared_cache_folder
|- z
  |- x
    |- y.png,
```

Листинг 7 — Структура кэша для модуля `mod_tile`

В листинге `'shared_cache_folder'` является названием общей папки, в которой хранится кэш, `'z'` — имя подпапки в `'shared_cache_folder'`, ассоциированное с уровнем зума карты, `'x'` — имя подпапки в `'z'`, ассоциированное с первым индексом плитки карты, а `'y'` — имя файла в `'x'`, ассоциированное со вторым индексом плитки карты (см рисунок 42).

Для получения плиток, составляющих «скользящую» карту, библиотека *Leaflet* генерирует запросы по шаблону:

```
http://path_on_server/{z}/{x}/{y}.png
```

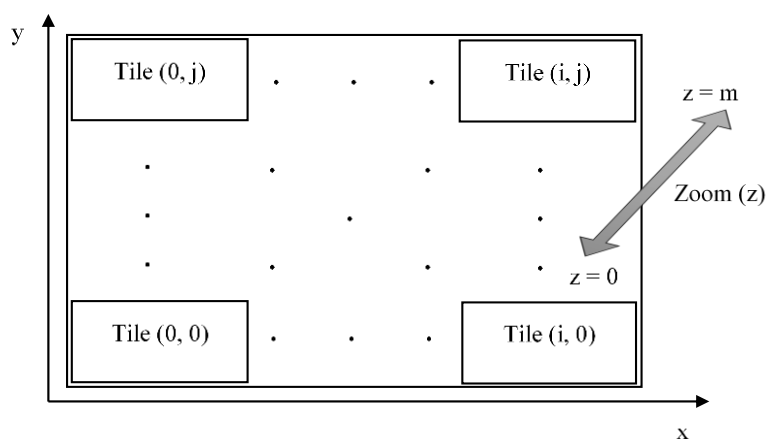


Рисунок 42 — Структура «скользящей» карты

При выполнении запроса в кэше производится поиск соответствующей плитки, которая отправляется клиенту. Если плитка отсутствует, то запускается модуль *renderd*, который создает ее и сохраняет в кэше.

В дополнение к программным интерфейсам *Leaflet*, было написано несколько функций с использованием языка программирования *JavaScript*, предоставляющих возможность отображения дополнительных слоев, содержащих пути, использующие один вид транспорта, и мультимодальные пути, точки интереса с описанием во всплывающем окне. Для мультимодальных путей дополнительно маркерами отмечаются точки пересадки и выводится всплывающее окно, содержащее информацию о пересадке (типы и маршруты транспортных средств, время их прибытия на остановку).

Сервис планирования маршрута был разработан на основе библиотеки *pgRouting*. Описание его реализации приведено в разделе 4.3.

Для предоставления возможности прямого и обратного геокодирования используется сервис *Nominatim*. Из-за того, что публичный сервис *Nominatim* имеет строгое ограничение на количество запросов, был сконфигурирован собственный сервис с использованием веб-сервера *Apache*. Запрос на получение координат по адресу (прямое геокодирование) выглядит следующим образом:

```
http://path_to_nominatim/search?format=json&q=queried+address
```

Данный запрос возвращает найденные места с координатами в формате *JSON*, которые обрабатываются устройством пользователя и отображаются на экране. Из нескольких предложенных вариантов пользователь может выбрать наиболее подходящий. В случае обратного геокодирования, когда пользователя отмечает место на карте, сервису *Nominatim* передаются координаты отметки, по которым осуществляется поиск объектов:

```
http://path_to_nominatim/reverse?format=json&lat=latitude&lon=longitude&zoom=zoom_level
```

Данный запрос возвращает адрес места, зависящий от уровня зума. На более высоких уровнях зума (при большем приближении) обратное геокодирование возвращает больше деталей. Например, на уровне зума 10 по координатам широты *lat=59.93404* и долготы *lon=30.30599*, результатом работы геокодера будет «Адмиралтейский район, Санкт-Петербург». По тем же координатам на уровне зума 10 результатом запроса будет «Исаакиевский собор»

### 4.3 Реализация сервиса формирования рекомендаций

Для ранжирования объектов, предоставляемых пользователю по его текущему местоположению, был разработан сервис формирования рекомендаций, основанный на модели, описанной в параграфе 3.4. При разработке сервиса использовался язык программирования *Python* и база данных под управлением *PostgreSQL*. В разделе представлено описание основных этапов работы сервиса.

Согласно модели формирования рекомендаций, сервис должен выполнять две основные функции:

- 1) Отслеживать появление оценок объектов в интеллектуальном пространстве и сохранять их вместе с контекстом, в котором они были выставлены.



- 2) Формировать предполагаемые оценки объектов, на основе которых будет осуществляться последующее ранжирование объектов в интерфейсе пользователя.

При взаимодействии с системой обеспечения инфомобильности, пользователи могут отдельно оценить объект, предложенный к ознакомлению, информацией о нем и соответствующие ему фотографии. При выставлении оценки объекта в интеллектуальное пространство отправляется триплет, содержащий оценку объекта и контекст пользователя:

```
([object_id], "tais:rate_object", RateObjectXML)

<?xml version="1.0" encoding="UTF-8"?>
<Evaluation xmlns="http://cais.iias.spb.su/XML/tais">
<UserID>[user_id]</UserID>
<Context>
<DateTime>2015-10-09 10:59 MSK</DateTime>
<Weather>Sunny</Weather>
<Company>Alone</Company>
</Context>
<Rating>4</Rating>
</Evaluation>
```

Листинг 8 — XML, содержащий оценку объекта с соответствующим контекстом

В структуре объекта *RateObjectXML* предоставляется описание контекста, привязанное к объекту и его оценке.

При оценке изображений и описания контекст не является необходимым, в связи с чем формируется следующие тройки, отправляемые в интеллектуальное пространство:

```
([description_url], "tais:rate_information", RateXML)
([image_url], "tais:rate_image", RateXML)

<?xml version="1.0" encoding="UTF-8"?>
<Evaluation xmlns="http://cais.iias.spb.su/XML/tais">
<UserId>[user_id]</UserId>
<Rating>4</Rating>
</Evaluation>
```

Листинг 9 — XML, содержащий оценку изображения или описания

Полученные оценки сохраняются в таблицах базы данных, созданной для хранения оценок пользователей. Структура БД представлена на рисунке 43. В БД определены следующие таблицы *AttractionRating*, *ImageRating*, *DescriptionRating* для хранения оценок достопримечательностей, изображений и описаний соответственно. При сохранении оценки объекта в дополнение к общим поля, содержащим системный идентификатор оцениваемого объекта (*object\_id*), идентификатор пользователя, оценившего объект, (*user\_id*) и выставленной оценки (*rating*), сохраняется контекст, в котором было совершено оценивание: время (*time*), компания (*company*) и погода (*weather*).

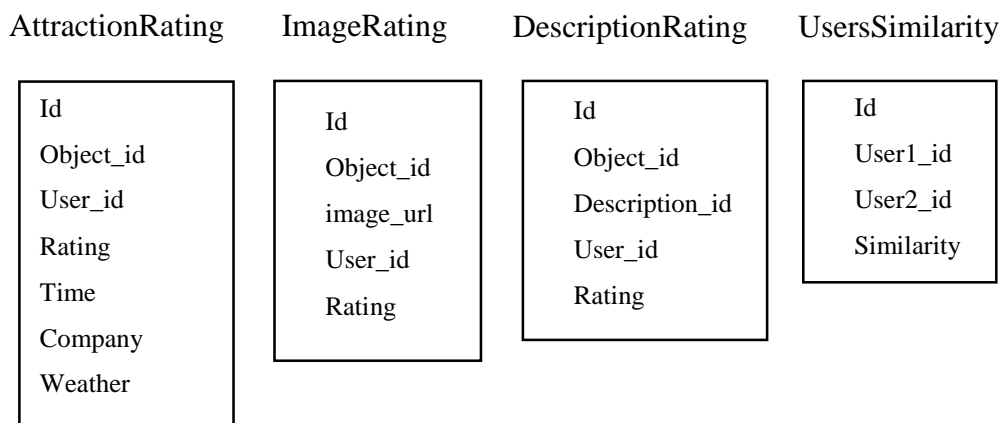


Рисунок 43 — Структура базы данных для хранения оценок пользователей

Как было описано в модели формирования рекомендаций, предполагаемые оценки формируются с помощью алгоритма коллаборативной фильтрации, в частности, метода, основанного на сходстве пользователей. Вычисление предполагаемых оценок начинается при обнаружении в интеллектуальном пространстве новых объектов, еще не имеющих оценок от пользователя, для которого они были определены. Обнаружение подобных объектов осуществляется с помощью операции подписки, предоставляемой платформой Smart-M3. При оценке достопримечательности клиентское приложение публикует в интеллектуальном пространстве следующий набор троек:

```
<mailto:example@example.com> ex:rated _:ratingNode .
_:ratingNode ex:attractionId <uid:9580> .
_:ratingNode ex:rating "5"^^xsd:integer .
```

Листинг 10 — RDF-тройки, соответствующие новой оценке

Здесь `<mailto:example@example.com>` — это URI, идентифицирующий пользователя системы, `<uid:9580>` — это URI, содержащий уникальный идентификатор объекта, присвоенный сервисом поиска объектов на карте, `_:ratingNode` обозначает анонимную вершину, соответствующую одной оценке объекта. Для получения информации, необходимой сервису формирования рекомендаций, необходимо оформить подписку на следующий *SPARQL*-запрос:

```
SELECT ?user_id ?att_id ?rating
{
  ?user_id ex:rated [
    ex:attractionId ?att_id ;
    ex:rating ?rating
  ].
}
```

Листинг 11 — SPARQL запрос на получение информации о географическом объекте в ИП

При обнаружении новых объектов в случае перемещения пользователя или подключении нового пользователя, сервис запускает вычисление предполагаемых оценок для объектов, еще не

оцененных пользователем. Оповещение о появлении новых объектов также реализуется с использованием механизма подписок в платформе *Smart-M3*. Подписка на появление новых объектов в интеллектуальном пространстве организуется следующим *SPARQL*-запросом:

```
SELECT ?user_id ?att_id ?rating
{
  ?user_id ex:nearBy [
    ex:attractionId ?att_id ;
  ].
}
```

Листинг 12 — *SPARQL* подписка на появление новых географических объектов в ИП

Вычисление предполагаемых оценок начинается с определения пользователей, имеющих похожие оценки по уже оцененным объектам относительно рассматриваемого пользователя. Вследствие того, что данная операция является одной из самых трудоемких, коэффициенты сходства пересчитываются в фоновом режиме и хранятся в соответствующей таблице в базе данных. Данная операция осуществляется триггером, который активируется при добавлении или изменении оценки объекта.

Расчет сходства между двумя пользователями с помощью вычисления косинусной меры реализован в функции, представленной в листинге 13.

```
def users_similarity(user1_rates, user2_rates):
    similarity = 0
    user1_normalize = normalize_ratings(user1_rates)
    user2_normalize = normalize_ratings(user2_rates)
    similarity = sum(map(lambda x, y: x * y if x is not None and
                        y is not None else 0, user1_normalize, user2_normalize)) / \
        sqrt(sum(map(lambda x: pow(x, 2) if x is not None else 0,
                    user1_normalize) *
              sum(map(lambda x: pow(x, 2) if x is not None else 0,
                    user2_normalize))))
    return similarity
```

Листинг 13 — Функция расчета сходства между пользователями

На вход функции вычисления коэффициента сходства пользователей передаются оценки объектов, которые к моменту расчета коэффициента были оценены пользователями. По имеющимся оценкам осуществляется процедура нормализации с помощью функции *normalize\_ratings*, представленной в листинге 14 и реализующей формулу (18). Функция принимает на вход массив оценок пользователя, отфильтровывает значения, соответствующие отсутствию оценки, вычисляет среднее значение, относительно которого будет проводиться нормализация и возвращает нормализованный список оценок. По нормализованным оценкам вычисляется степень сходства оценок пользователей с применением косинусной меры (19). Результат вычисления заносится в соответствующее поле таблицы *Similarity*.

```
def normalize_ratings(user_rates):
    user_filtered = filter(lambda x: x is not None, user_rates)
    user_sum_rating = sum(user_filtered)
    user1_length = len(user_filtered)
    return map(lambda x: x - ((3 + user_sum_rating) /
        (user1_length + 1)) if x is not None else None, user_rates)
```

Листинг 14 — Функция нормализации оценок

Для последующего вычисления предполагаемой оценки выбираются пользователи, для которых степень сходства оценок с оценками текущего пользователя превышает 0.5. Вычисление предполагаемой оценки производится по формуле (20), реализованной в функции *calculate\_rate()*, представленной в листинге 15. На вход функции передается идентификатор объекта, для которого необходимо вычислить предполагаемую оценку, идентификатор пользователя, которому будет отправлен результат вычисления, список пользователей со схожими интересами и пороговое значение сходства. Вычисление предполагаемого значения также осуществляется с использованием нормализованных оценок пользователей.

```
def calculate_rate(object_id, current_user, similar_users, threshold):
    rate = 0
    normalized_rates = {}
    for user in similar_users:
        normalized_rates[user] = normalize_ratings(rates[user])
    rate = sum(map(lambda x: normalized_rates[x][object_id] *
        similarity_matrix[current_user][x] if
        similarity_matrix[current_user][x] > threshold and
        normalized_rates[x][object_id] is not None else 0,
        similar_users)) / \
        sum(map(abs, filter(lambda x: x > threshold,
        similarity_matrix[current_user].values()))))
```

Листинг 15 — Функция расчета предполагаемой оценки объекта

Предполагаемых оценок для еще не оцененных объектов объединяются с уже выставленными оценками и список объектов сортируется по убыванию оценок. Отсортированный список выкладывается в интеллектуальное пространство, откуда его загружает приложение пользователя.

Тестирование сервиса формирования рекомендаций осуществлялось на том же тестовом стенде, что и сервиса планирования маршрутов. Для тестирования генерировались случайные пользователи, объекты и соответствующие случайные оценки для них. Количество пользователей увеличивалось от 100 до 1000 человек с шагом в 100 и для каждой группы количество оцениваемых объектов также изменялось от 100 до 1000 с шагом 100. При тестировании сервиса формирования рекомендаций были получены следующие результаты. Наиболее затратной операцией с точки зрения использования ресурсов является расчет меры

сходства между пользователями. При этом увеличение затрат зависит только от количества пользователей, что отражено на рисунке 44.

При вычислении предполагаемой оценки время вычисления зависит от количества пользователей и объектов, а также от распределения оценок пользователей, влияющего на формирование групп пользователей с совпадающими оценками. Зависимость времени расчета от количества пользователей и объектов представлена на рисунке 45.

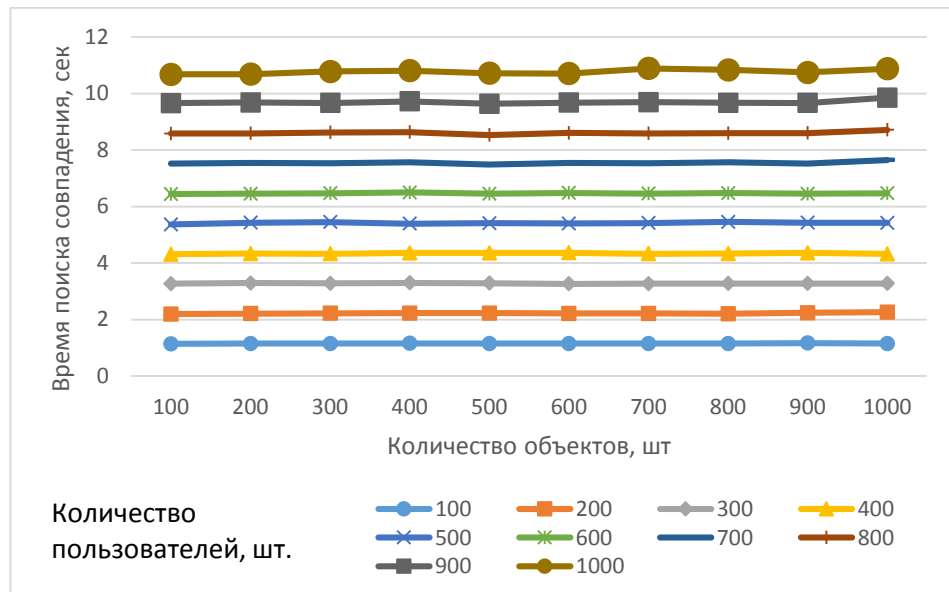


Рисунок 44 — Зависимость времени вычисления коэффициентов сходства между пользователями от количества пользователей и оцениваемых объектов

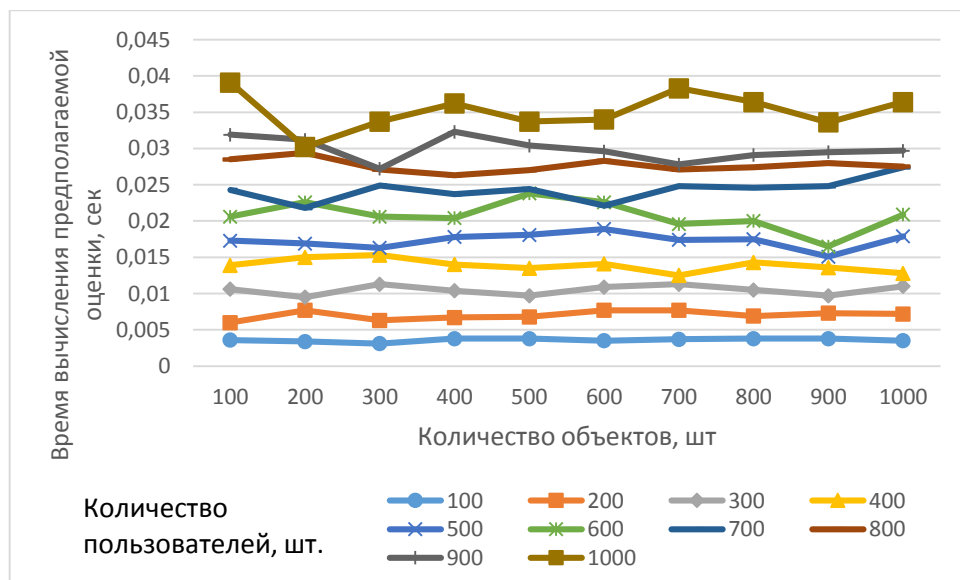


Рисунок 45 — Зависимость времени вычисления предполагаемой оценки объекта от количества пользователей и оцениваемых объектов

## 4.4 Реализация сервиса сопоставления онтологий

### Автоматическое сопоставление онтологий

Сервис сопоставления онтологий был реализован с использованием высокоуровневого языка программирования *Python*. Согласно описанию метода, алгоритм сопоставления должен выполнить три основные операции: лингвистическое сравнение термов, представляющих классы онтологии, включающее в себя их синтаксическое и семантическое сравнение; пересчет метрик на графе, представляющем иерархию онтологии. Каждая из операций представлена в сервисе отдельной функцией, описание которых представлено далее.

На лингвистическом уровне сопоставление классов онтологии производится попарно. На вход функции, отвечающей за лингвистический анализ термов онтологии, подается пара, состоящая из термов двух сопоставляемых онтологий. Результатом работы данной функции является числовая метрика, показывающая, насколько похожи термы онтологий.

Для синтаксического сравнения вычисляется числовая метрика с применением нечеткого сопоставления строк. Реализация нечеткого сопоставления строк на языке *Python* возможна как с использованием встроенных функций языка, так и с использованием расширений на языке C, которые подключаются к программе на *Python* с помощью специальной оболочки.

Встроенным средством, позволяющим сравнивать строки в *Python*, является класс *SequenceMatcher* библиотеки *difflib* [67]. Для сравнения используются следующие функции:

- 1) *ratio()*. Разбивает строки на подмножества, среди которых проводит сравнение.

При вычислении степени сходства используется следующая метрика:

$$w(s_1, s_2) = \frac{2 \cdot |s_1 \cap s_2|}{|s_1| + |s_2|},$$

где  $|s_1|, |s_2|$  — количество подмножеств в строковых последовательностях,  $|s_1 \cap s_2|$  — количество совпадающих подмножеств последовательностей. Степень сходства находится в диапазоне  $w(s_1, s_2) \in [0, 1]$ .

- 2) *quick\_ratio()* — функция, позволяющая быстро оценить степень сходства. Производит посимвольное сравнение и возвращает верхнюю границу возможной степени сходства.

Для языка *Python* разработаны библиотеки, позволяющие находить различные метрики сходства строк. Наиболее часто применяемой метрикой является расстояние Левенштейна [13]. Данная метрика оценивает число операций редактирования строки (вставка, удаление, замена символа), необходимых для преобразования одной строки в другую. Похожей метрикой, более подходящей для коротких строк, является расстояние Джаро-Уинклера (*Jaro-Winkler*

*distance*) [176]. Обе метрики представлены в модуле *python-Levenshtein* [108], предоставляющем следующие функции анализа строк, позволяющие оценить степень их сходства:

- *distance(string1, string2)*. Вычисление абсолютного расстояния Левенштейна;
- *hamming(string1, string2)*. Вычисление расстояния Хемминга. Редко применяется при определении сходства строк, поскольку подходит только для строк равной длины;
- *jaro\_winkler(string1, string2[, prefix\_weight])*. Вычисление метрики сходства на основе расстояния Джаро-Уинклера. Результат вычисления находится в интервале [0,1];
- *ratio(string1, string2)*. Вычисление метрики сходства на основе расстояния Левенштейна. Результат вычисления также находится в интервале [0,1];
- *segratio(string\_sequence1, string\_sequence2)*. Вычисление сходства между множествами строк.

Дополнительным преимуществом использования библиотеки является поддержка кодировки Unicode, что позволяет оценивать сходство строк в любых языках.

Комбинация расчета метрики с использованием стандартного модуля Python и расстояния Левенштейна представлена в модуле *fuzzywuzzy* [77]. Функции данного модуля позволяют получать следующие метрики сходства строк:

- *fuzz.ratio(string1, string2)* Простой рейтинг, использующий расстояние Левенштейна и метрику из стандартной библиотеки Python.
- *Fuzz.partial\_ratio(string1, string2)* Частичный рейтинг. Возвращает
- *fuzz.token\_sort\_ratio(string1, string2)*. Рейтинг использующий сортировку лексем. При использовании данной функции строка разбивается на лексем, которые сравниваются после сортировки.
- *fuzz.token\_set\_ratio(string1, string2)*. В данной функции лексем после сортировки объединяются в множества совпадающих лексем, и сравнение идет между множествами.

Все метрики, возвращаемые библиотекой, расположены в интервале от 0 до 100. Сравнение метрик для термов, '*DPSTART*' и '*START\_DR*', представляющих начальную точку пути водителя в разных онтологиях, представлено на рисунке 46. Чем ближе каждая метрика к значению «1», тем более похожими считаются термы. Значение, равное «1», следует интерпретировать как полное совпадение термов, а «0» — как абсолютное различие.

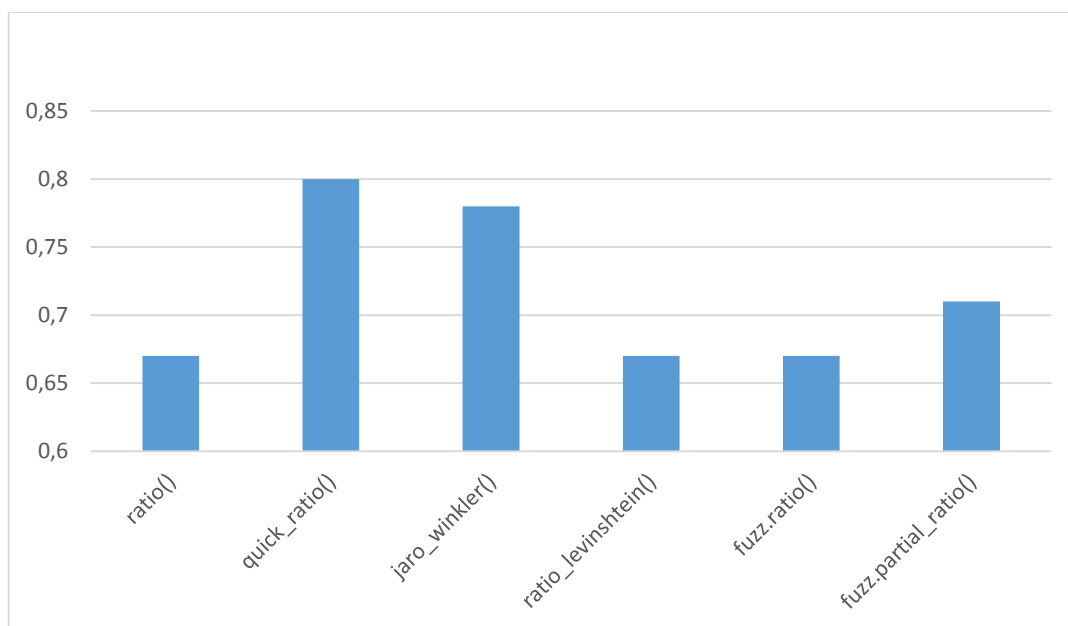


Рисунок 46 — Сравнение метрик, полученных функциями сравнения строк

Как видно из рисунка 46, самое высокое значение сходства было вычислено при быстром анализе строк функцией *quick\_ratio()* из стандартной библиотеки *Python*. Метрика, полученная функцией *jaro\_winkler()* близка к метрике быстрого анализа по причине того, что для перехода от строки *'DPSTART'* к строке *'START\_DR'* требуется незначительное число перестановок. Метрика, полученная при вычислении частичного совпадения строк, также превосходит остальные, в связи с тем, что термы имеют значительную общую часть *'START'*. Для семантического анализа строк в сервисе сопоставления онтологий выбрана функция *fuzz\_ratio* библиотеки *fuzzywuzzy* вследствие того, что она учитывает расстояние Левенштейна вместе со сравнением подмножеств, что позволяет вычислить более точную метрику совпадения строк.

Для проведения следующего шага лингвистического уровня — семантического сопоставления термов — осуществляется обращение к словарю *WordNet*. Запрос позволяет получить список связанных слов для каждого из термов. Среди связанных слов выделяются синонимы, ассоциированные и совпадающие слова. Использование словарей, однако, накладывает определенные ограничения на выбор термов — они должны быть словами естественного языка. В противном случае обращение к словарям будет не будет возвращать результат.

Для работы с *WordNet* используется набор приложений для работы с естественным языком *Natural Language Toolkit NLTK* [47]. Данный набор позволяет осуществлять поиск определений слова, синонимов, антонимов, гиперонимов, и других связанных понятий, которые используются для оценки семантического расстояния в методе сопоставления онтологий. Также, с помощью данного набора можно произвести семантическую оценку близости слов, основанную на графе отношений между понятиями, существующем в *WordNet*.



Процесс вычисления семантического расстояния для термов представлен в листинге на примере термов, задающих тип транспортного средства «car» и «automobile»

```
s1 = 'car'
s2 = 'automobile'
car = wn.synsets(s1)
automob = wn.synsets(s2)
count = []
for e1 in car:
    mult_w = 1
    for e2 in automob:
        similarity = e1.path_similarity(e2)
        if similarity == 1.0:
            count.append(float("inf"))
        if 1.0 > similarity >= 0.5:
            count.append(0.5)
        else:
            count.append(0.3)
        print e1.path_similarity(e2)
    count.append(mult_w)
print count
print 1/sum(count)
```

Листинг 16 — Вычисление семантического расстояния в базе WordNet

Для вычисления семантического расстояния производится поиск возможных определений для термов с помощью функции *synsets(term)*. Данная функция возвращает список всех возможных семантических значений аргумента, которые также являются его синонимами.

Функция *path\_similarity()* позволяет найти расстояние между термами в графе связей *WordNet*. Чем меньше понятий связывает термы, тем ближе они находятся друг у другу. Для совпадающих термов расстояние равняется «1.0» и чем дальше они друг от друга, тем ближе расстояние к «0.0».

В ходе перебора термов онтологий составляется матрица расстояний между парами термов. Дальнейший пересчет матрицы осуществляется с использованием структуры онтологии, согласно формулам, представленным в 3.2.

### **Создание задач на сопоставление онтологий для краудсорсинговой платформы Amazon Mechanical Turk**

Для реализации ручного сопоставления онтологий используются возможности краудсорсинговой платформы *Amazon Mechanical Turk (AMT)*. Данная платформа объединяет более полумиллиона пользователей из 190 стран, способных решать задачи различной степени сложности. В настоящий момент с помощью платформы решается более 190000 различных заданий, связанных с различными видами классификации информации, распознаванием изображений, проведением опросов и т. п.

Amazon предоставляет возможность создания заданий через веб-интерфейс с последующим размещением в списке всех задач проекта, либо автоматическое формирование заданий с применением специального программного интерфейса для доступа к платформе. Для отладки создана «песочница», в которой разработчики могут проверить корректность процесса формирования задачи, то, как она будет выглядеть для пользователя и автоматизацию сбора решений. Для доступа к песочнице необходима регистрация, в ходе которой формируется идентификатор пользователя и секретный ключ авторизации. Использование песочницы для формирования задачи на сопоставление онтологий представлено далее.

Возможность разработки заданий для платформы *Amazon Mechanical Turk* предоставляется для наиболее популярных языков программирования. Официальные наборы для разработки приложений (*software development kit* — *SDK*) доступны для языков, используемых при разработке приложений, работающих с сетью Интернет, в число которых входят *Java*, *Perl*, *.NET*, *Rubi*. Для разработки на языке *Python* существует неофициальная библиотека *Boto* [79], позволяющая получить доступ к основным сервисам Amazon, в число которых входит платформа *Mechanical Turk*.

Процесс формирования задачи *human intelligence task (HIT)* с использованием библиотеки *Boto* представлен в листинге 17.

```

1. mtc = MTurkConnection(aws_access_key_id=ACCESS_ID,
                        aws_secret_access_key=SECRET_KEY,
                        host=HOST)

2. t1 = 'car'
3. t2 = 'automobile'

4. title = 'Match two terms' // Заголовок задачи
5. description = ('Please, set the matching degree of the following'
                 ' ontology classes') // Описание задачи
6. keywords = 'ontology, matching' //Ключевые слова для поиска

7. ratings = [('The same', 0.0),
              ('Synonyms', 0.5),
              ('Different', 1.0)] //Варианты ответов на задачу

#----- Создание обзорной страницы HIT -----

8. overview = Overview()
9. overview.append_field('Title', 'Set the matching degree of the
                          following terms:')
10.overview.append(FormattedContent(t1))
11.overview.append(FormattedContent(t2))

#----- Формирование вопроса для исполнителей -----

```

```

12. qc1 = QuestionContent()

13. ftal = SelectionAnswer(min=1, max=1, style='dropdown',
                           selections=ratings,
                           type='text',
                           other=False)

14. q1 = Question(identifier='matching',
                  content=qc1,
                  answer_spec=AnswerSpecification(ftal),
                  is_required=True)

#----- Форматирование вопроса -----

15. question_form = QuestionForm()
16. question_form.append(overview)
17. question_form.append(q1)

#----- Создание задачи HIT -----

18. mtc.create_hit(questions=question_form,
                  max_assignments=1,
                  title=title,
                  description=description,
                  keywords=keywords,
                  duration=60*5,
                  reward=0.00,
                  response_groups=['Minimal'])

19. print hit[0].HITId

```

Листинг 17 — Продолжение

В задаче исполнителям предлагается оценить, насколько понятия онтологий сопоставимы друг с другом. Для этого предоставляются термины онтологии и возможность выбора степени совпадения их трех значений: «полное совпадение», «сходные понятия» (синонимы), «различающиеся». При создании задачи также указываются ключевые слова, по которым ее можно найти, рекомендуемое время выполнения и вознаграждение, выплачиваемое за решение задачи. В связи с невысокой сложностью задачи сопоставления понятий онтологий плата за ее выполнение не предусмотрена. Однако пользователи, успешно выполнившие задание, получают дополнительную квалификацию, что, согласно правилам платформы, позволяет им претендовать на более сложные и высокооплачиваемые задания.

## HITs containing 'match ontolog'

1-1 of 1 Results

Sort by: HITs Available (most first) 60

[Show all details](#) | [Hide all details](#)

Match two terms		<a href="#">View a HIT in this group</a>	
Requester: <a href="#">Nikolai</a>	HIT Expiration Date: Aug 30, 2015 (6 days 23 hours)	Reward: \$0.00	
	Time Allotted: 5 minutes	HITs Available: 2	

Рисунок 47 — Обзор задачи HIT для платформы AMT

Timer: 00:00:14 of 5 minutes

Finished with this HIT? Let someone else do it?

[Submit HIT](#)[Return HIT](#)Total Earned: \$0.00  
Total HITs Submitted: 2 Automatically accept the next HIT

Match two terms			
Requester: <a href="#">Nikolai</a>	Reward: \$0.00 per HIT	HITs Available: 1	Duration: 5 minutes
Qualifications Required: None			

## Set the matching degree of the following terms:

car  
automobile

## Select:

[The same](#)

Finished with this HIT? Let someone else do it?

[Submit HIT](#)[Return HIT](#) Automatically accept the next HIT

Рисунок 48 — Процесс решения задачи HIT

Список задач, размещенных на платформе может быть получен функцией `get_all_hits()`. Для каждой задачи дополнительно предоставляется ее статус: “Reviewable”, для задач, имеющих минимум одно решение, и “Assignable” для задач, еще не имеющих ни одного решения. Задача считается полностью решенной, после получения 10 ответов на нее. Ответы загружаются для обработки с помощью функции, представленной в листинге 18.

```
def get_all_reviewable_hits(mtc):
    p_size = 50
    hits = mtc.get_reviewable_hits(page_size=p_size)
    print "Total results to fetch %s " % hits.TotalNumResults
    print "Request hits page %i" % 1
    total_pages = float(hits.TotalNumResults)/p_size
    int_total= int(total_pages)
    if(total_pages-int_total>0):
        total_pages = int_total+1
    else:
        total_pages = int_total
    pn = 1
    while pn < total_pages:
        pn = pn + 1
        print "Request hits page %i" % pn
        temp_hits =
    mtc.get_reviewable_hits(page_size=p_size,page_number=pn)
```

Листинг 18 — Получение ответов от исполнителей

```

hits.extend(temp_hits)

return hits

mtc = MTurkConnection(aws_access_key_id=ACCESS_ID,
                      aws_secret_access_key=SECRET_KEY,
                      host=HOST)

hits = get_all_reviewable_hits(mtc)

for hit in hits:
    print "HIT ID: %s" % hit.HITId
    assignments = mtc.get_assignments(hit.HITId)
    for assignment in assignments:
        print "Answers of the worker %s" % assignment.WorkerId
        for question_form_answer in assignment.answers[0]:
            print question_form_answer.fields
        print "-----"

#>>> // Пример ответа
Total results to fetch 2
Request hits page 1
HIT ID: 3OPLMF3EU5GPM9KUH8AGP17J5XFLN2 // Идентификатор задачи
Answers of the worker AC5JPOFDYZ1XG // Идентификатор исполнителя
[u'0.0'] // Ответ исполнителя
-----
HIT ID: 34F34TZU7WSN2X71S40JHYRTQDJJ27
Answers of the worker AC5JPOFDYZ1XG
[u'0.5']
-----

```

## Листинг 18 — Продолжение

С учетом того, что задача имеет крайний срок исполнения, требуется контроль даты завершения задачи и ее продление, в случае превышения необходимого срока решения. После получения необходимого количества ответов и извлечения их из платформы с помощью функции, представленной в листинге 18, задача отмечается как выполненная и удаляется с помощью вызова функции `disable_hit(HIT_ID)` класса `MTurkConnection`. Ответы обрабатываются, согласно формуле (10) и вносятся в матрицу  $M$ , в позицию, соответствующую рассматриваемым понятиям онтологии.

Для проверки метода сопоставления онтологий были выбраны две онтологии для сервиса поиска объектов на основе местоположения пользователя, полученные от разных разработчиков. При проведении визуального сравнения были обнаружены следующие различия: различались классы `datetime` и `timestamp`; `Latitude` и `y`; `Longitude`, `x`; `Image`, `Photo`; `File`, `Link`; а также, в одной из онтологий отсутствовали классы `Context` и `Rating` вместе со связанными атрибутами.

Для оценки оперативности работы с краудсорсинговой платформой и точности полученных результатов, было проведено полное сопоставление полученных онтологий,

состоящих из 16 и 13 классов соответственно, исключительно силами интернет-сообщества. Для этого в АМТ было создано 208 микрозаданий с возможностью назначения пяти исполнителей на каждое. В решении задач приняло участие 36 исполнителей. За неделю, в течении которой проводилось тестирование, задания были решены 791 раз (включая повторы), что в среднем составляет около 22 заданий на исполнителя. В среднем, на решение одного микрозадания тратилось около 93 секунд, при допустимом времени на размышление, равном пяти минутам.

При запуске полной версии метода сопоставления онтологий для классов онтологий, которые при визуальном сопоставлении полностью совпали, было также найдено однозначное совпадение: расстояние между классами равнялось «1,0», что является максимально возможным значением и явно указывает на полное соответствие данных частей онтологии. Также, совпадения выше были обнаружены для классов  $dist(Image, Photo) = 0,76$  и  $dist(File, Link) = 0,81$  при значении порогового коэффициента  $Tr = 0,75$ . Для оставшихся классов проводилось сравнение с результатами решения краудсорсинговых микрозаданий, результаты которого представлены в таблице 6.

Таблица 6 — Сравнение результатов автоматического сопоставления и сопоставления с использованием технологии краудсорсинга

Сравниваемые классы	Результат автоматического сопоставления	Результат краудсорсинга
datetime -> timestamp	0,52	0,9
Latitude -> y	0,12	0,8
Longitude -> x	0,09	0,8

Получение подобных результатов может быть обосновано использованием термов, не полностью представленных в словарях, но имеющих смысл для человека. Таким образом, использование технологии краудсорсинга совместно с автоматическим сопоставлением онтологий позволяет повысить полноту сопоставления за счет использования коллективного сознания и опыта.

#### 4.5 Реализация сервиса обеспечения конфиденциальности информации о пользователе

При разработке сервиса для обеспечения конфиденциальности информации о пользователе для системы обеспечения инфомобильности применялся язык программирования Python. Его выбор обоснован мощными встроенными инструментами для работы с множествами и логическими утверждениями, а также наличием большого количества программных модулей, с помощью которых можно значительно расширить возможности разработки приложений.

Основная функциональность сервиса реализована в трех главных функциях, соответствующих типам правил, описанных в 3.5. Первая функция, “formalize\_context”, собирает контекст пользователя и преобразует его в уровни доверия согласно имеющимся экспертным оценкам для каждого возможного значения атрибутов. Оценки для каждого сервиса, использующего модуль контроля и разграничения доступа задаются с помощью специальной утилиты для настройки модуля и хранятся в отдельной базе данных брокера безопасности в виде пары «ключ:значение», которая в языке Python представлена типом dictionary - {key1:value1, key2:value2, ..., keyN:valueN}

```
SOCIAL_NETWORK = {"012345": ["543210", "897234", "098324"],
                  "543210": ["012345", "283675", "098324"]}

association_rules = {"friendship": {"friend": 0.9, "not_friend": 0.1},
                    "is_a": {"Driver": 0.7, "Passenger": 0.6,
                              "Service": 0.8}
                    }

ranges_for_roles = {"service_role": {"friendship": [0.8, 1],
                                     "is_a": [0.5, 1]},
                   "friend_role": {"friendship": [0, 0.79],
                                    "is_a": [0, 0.49]},
                   "non_friend_role": {"friendship": [0, 1],
                                       "is_a": [0, 1]}
                   }

role_actions = {"service_role": ["read_all_inf", "", "", ],
                "friend_role": ["read_only_public"],
                "non_friend_role": ["read_only_public"]}
}
```

Листинг 19 — Пример соответствия атрибутов класса уровням доверия

```
def formalize_context(participaint_inf_answer,
                      participaint_inf_request, association_rules):
    context = {}
    for key in association_rules:
        if key == "friendship": # Special feature
            if participaint_inf_request["social_network_id"] in \
                SOCIAL_NETWORK[participaint_inf_answer["social_network_id"]]:
                context["friendship"] = \
                    association_rules["friendship"]["friend"]
            else:
                context["friendship"] =
                    association_rules["friendship"]["not_friend"]
        else:
            context[key] = \
                association_rules[key][participaint_inf_answer[key]]
    return context
```

Листинг 20 — Функция вычисления уровня доверия для атрибутов контекста

Вторая функция, “set\_role”, использует результаты работы первой, и набор правил, характеризующих необходимые значения уровней доверия для компонент контекста, для определения ролей пользователю системы совместного использования автотранспорта. Согласно разделу 3.5 для этой цели используется логическая функция произведения соответствия уровней доверия компонент контекста, определенных первым типом правил, заданным интервалам, соответствующим какой-либо роли. Если все произведение в итоге дает значение «Истина», то пользователю присваивается роль, для которой проверялся набор правил. Подобным образом из всех возможных ролей для пользователя выбираются только те, которые удовлетворяют всем ограничениям.

```
def set_role(user_context, ranges_for_roles):
    user_roles = []

    for key in ranges_for_roles:
        OverallFlag = True
        for key2 in ranges_for_roles[key]:
            OverallFlag = OverallFlag and \
                (ranges_for_roles[key][key2][0] < user_context[key2]) and \
                (ranges_for_roles[key][key2][1] > user_context[key2])
        if OverallFlag:
            user_roles.append(key)
    return user_roles
```

Листинг 21 — Функция определения роли

Для сервиса обеспечения безопасности конфиденциальных данных были введены следующие роли: *service\_role*, *friend\_role*, *non\_friend\_role*. Роль *service\_role* присваивается только сервисам, полностью удовлетворяющим параметрам контекста, заданным пользователями. Роль *friend\_role* позволяет предоставить доступ к конфиденциальным данным другим пользователям, состоящим в дружеских отношениях с пользователем. Роль *non\_friend\_role* присваивается всем участникам ИП, не попадающим в требуемые параметры контекста. Параметры контекста, требуемые для присвоения каждой из ролей, определяются и хранятся на устройствах пользователей. Третья функция проверяет разрешения, принадлежащие роли, которая присвоена участнику, и сравнивает их с разрешениями, необходимыми для доступа к запрашиваемой информации. Если они совпадают, то доступ считается разрешенным и участники получают оповещение, позволяющее предоставить доступ к запрашиваемому ресурсу. В противном случае пользователи уведомляются об обратном.

При получении разрешения на передачу информации, участник, предоставляющий ее, зашифровывает необходимую информацию с помощью открытого ключа участника, запрашивающего информацию, и размещает шифр вместе с подписью в интеллектуальном пространстве. Используя свой закрытый ключ участник ИП, получивший оповещение о передаче



информации, получает шифр, расшифровывает его и верифицирует цифровую подпись. После прохождения верификации информация передается на обработку.

```
def set_actions(user_role, role_actions):
    allowed_actions = []
    for role in user_role:
        allowed_actions = list(set(allowed_actions) |
                                set(role_actions[role]))
    return allowed_actions
```

Листинг 22 — Функция выдачи разрешения на доступ к информации

Шифрование информации реализовано при помощи алгоритма асимметричного шифрования RSA. Для языка Python разработан модуль `rsa` [139], соответствующий стандарту криптографии с открытым ключом PKSC #1 [148] и позволяющий встроить необходимые функции шифрования в программу.

Модуль `rsa` поддерживает следующие функции, требуемые для реализации шифрования в системе обеспечения инфомобильности:

- Генерация ключа. Поддерживается возможность генерации ключей длиной до 4096 бит. Для системы обеспечения инфомобильности была выбрана длина ключа в 2048 бит, позволяющая достичь достаточной стойкости шифра к существующим атакам;
- Шифрование. Операция шифрования осуществляется с использованием открытого ключа. Также поддерживается шифрация сообщений, имеющих длину больше, чем ключ за счет разбиения на блоки, не превышающие по размеру ключ, которые шифруются независимо друг от друга и затем объединяются в одну криптограмму;
- Дешифрование. Процесс идет обратно шифрованию. Криптограмма разбивается на блоки, не превышающие по размеру ключ и дешифруются с использованием закрытого ключа. Расшифрованные блоки затем объединяются в одно сообщение;
- Формирование цифровой подписи. Для формирования цифровой подписи вычисляется контрольная сумма с использованием алгоритмов ‘MD5’, ‘SHA-1’, ‘SHA-256’, ‘SHA-384’ или ‘SHA-512’, которая затем шифруется при помощи закрытого ключа. В системе обеспечения инфомобильности для вычисления ключевой суммы используется алгоритм SHA-1;
- Верификация цифровой подписи. Для полученного сообщения вычисляется контрольная сумма, которая сравнивается с результатом дешифрования подписи. При их совпадении подпись считается верифицированной и сообщение — заслуживающим доверия.

Таким образом, при возникновении необходимости передачи конфиденциальной информации в интеллектуальном пространстве, происходит обмен следующей информацией.

Сервис интеллектуального пространства отправляет запрос (в форме RDF) на доступ к некоторой информации, в публичное интеллектуальное пространство и отслеживает получение к ней доступа:

```
service1.smart_space.insert("service1_ID", "request", "resource");
service1.smart_space.subscribe("service1_ID", "access_granted", None);
```

Сервис интеллектуального пространства, предоставляющий запрашиваемую участником информацию, принимает его запрос и запрашивает у брокера безопасности разрешение на доступ данного участника к запрашиваемой информации.

```
service2.smart_spase.insert("service2_ID", "participant _requested",
"user_ID");
service2.smart_spase.insert("service2_ID", "resource_type", "type");
```

Брокер безопасности считывает контекст участника и проверяет его цифровую подпись, используя соответствующий открытый ключ. Если подпись проходит проверку, то брокер считает сервис аутентифицированным и применяет правила, закрепленные в политике безопасности, для назначения ему роли в интеллектуальном пространстве. Разрешение на доступ к ресурсу формируется исходя из набора разрешений, закрепленных за назначенной ролью. Ответ, содержащий запрет или разрешение подписывается закрытым ключом брокера и отправляется сервису, который запросил права доступа для пользователя.

```
security_broker.smart_space.insert("Security Broker", "participant",
"participant_ID");
security_broker.smart_space.insert("Security Broker", "access", "granted"
or "denied");
```

Сервис, у которого участник запрашивает информацию, получает ответ от брокера и, если доступ к информации разрешен, создает виртуальное приватное интеллектуальное пространство. Для этого он собирает информацию, запрошенную сервисом, шифрует ее с использованием открытого ключа сервиса, вычисляет контрольную сумму исходного сообщения и формирует цифровую подпись с использованием своего закрытого ключа. Шифр вместе с подписью размещается в интеллектуальном пространстве:

```
service2.smart_space.insert("service1_ID", "access_granted", "True");
service2.smart_space.insert("service1_ID", "private_space",
Encrypted_message);
service2.smart_space.insert("service1_ID", "signature", Encrypted_hash);
```

Если доступ был запрещен, сервису отправляется соответствующее уведомление:

```
service.smart_space.insert("participant_ID", "access_granted", "FALSE");
```

Сервис, который отправлял запрос на получение информации, получает уведомление по подписке в интеллектуальном пространстве. Если доступ был получен, и в ИП появилось зашифрованное сообщение с подписью, сервисом совершается загрузка сообщения и его проверка. После окончания передачи информации, виртуальное приватное пространство удаляется.

Оценка работы модуля обеспечения конфиденциальности информации пользователей системы обеспечения инфомобильности производилась по следующим основным параметрам:

- Время отклика системы. Под этим параметром подразумевается суммарное время, затрачиваемое системой, начиная от момента отправки запроса пользователем и заканчивая ответом сервиса и получением информации.
- Требуемый объем оперативной памяти. Данный параметр показывает суммарные затраты оперативной памяти на одном устройстве пользователя и одном информационном брокере.
- Объем программного кода. Данный параметр показывает объем памяти на постоянном накопителе устройства, необходимый для установки программного обеспечения для работы сервиса одном устройстве пользователя и одном информационном брокере.
- Нагрузка на сеть. Данный параметр показывает число обращений к ИП через протокол SSAP за время обработки запроса.

Результаты испытаний приведены в таблице 7.

Таблица 7 — Сравнение основных системных требований системы обеспечения инфомобильности без модуля обеспечения конфиденциальности и с ним

Параметр	Система без модуля контроля и разграничения доступа	Система с модулем контроля и разграничения доступа	Процент прироста
Время отклика системы	110 мс	130 мс	18%
Используемый объем оперативной памяти	Инф. брокер – 14 Мб Клиентское ПО – 9.7 Мб	Инф. брокер – 14 Мб Клиентское ПО – 10.8 Мб Брокер безопасности - 4.5 Мб	24%
Объем программного кода	Инф. брокер – 74 Кб Клиентское ПО – 481 Кб	Инф. брокер – 74 Кб Клиентское ПО – 485 Кб Брокер безопасности - 11 Кб	3%
Нагрузка на сеть	55 запросов от инф. брокера 38 запросов от клиента	55 запросов от инф. брокера 42 запроса от клиента 3 запроса от брокера безопасности	8%

#### **4.6 Апробация сервис-ориентированной системы обеспечения инфомобильности в области туризма**

Разработанная архитектура и сервисы были использованы при разработке систем обеспечения инфомобильности в области туризма для республики Карелия и города Санкт-Петербурга.

Для республики Карелия разработка велась в рамках проекта КА322 «Разработка трансграничной туристической информационной инфраструктуры (Smart e-Tourism)» (Development of cross-border e-tourism framework for the programme region) [64], реализуемой с 2012 по 2014 годы, программы приграничного сотрудничества в рамках европейского инструмента соседства и партнёрства «Карелия» (ENPI) реализуемой на территории Финляндии (Регион Кайну, Северная Карелия, Регион Оулу) и России (Республика Карелия) [101].

Целью проекта КА322 являлась разработка мобильных сервисов для оказания поддержки туристам при путешествии по целевому региону программы ENPI. Под оказанием поддержки подразумевается предложение информации туристической направленности, такой как основные достопримечательности региона с описанием и фотографиями, способы перемещения по региону, поиск и бронирование гостиниц и другие. Одним из условий оказания поддержки был учет потребностям и индивидуальных предпочтений туриста:

Результаты, полученные в диссертационной работе, были успешно применены для разработки интеллектуального гида для туристов в рамках программы ENPI. Интеллектуальный гид обеспечивает инфомобильность туриста при посещении различных регионов земного шара. Основной функциональностью интеллектуального гида является:

- Поиск достопримечательностей. С помощью сервиса поиска объектов из геоинформационных систем выбираются объекты, отмеченные как достопримечательности: музеи, памятники, мосты и другие. Поиск осуществляется как по текущему местоположению, так и по указанному пользователем вручную.
- Поиск информации о достопримечательности. С помощью сервиса поиска информации осуществляется поиск информации, привязанной к найденным объектам. Информация включает в себя описание достопримечательности и ее фотографии.
- Рекомендации достопримечательностей. С помощью сервиса формирования рекомендаций осуществляется анализ предпочтений пользователя, на основе которого найденные достопримечательности ранжируются по степени убывания их интереса для пользователя. Таким же образом осуществляется ранжирование фотографий. Также пользователя могут выставить свои оценки для предложенных

объектов и фотографий, повысив, тем самым, точность последующих рекомендаций.

- Поиск способов перемещения между достопримечательностями. При выборе способов передвижения учитываются виды общественного транспорта, доступные в регионе, а также предлагается совершать совместные поездки с использованием личного автомобильного транспорта. Для общественного транспорта имеется возможность построения мультимодальных маршрутов. Расписание движения ОТ для республики Карелия предоставлено компанией Яндекс для маршрутов в г. Петрозаводске и сервисом Яндекс.Расписания для междугородних маршрутов. В дополнение к общественному и личному транспорту предлагается совместное использование автотранспорта, которое направлено на внедрение нового способа недорогого путешествия в регионе, а также увеличение туристической доступности региона. Сервис позволяет туристам найти водителей, желающих разделить поездку с туристами. Интеграция сервиса с социальными сетями позволяет туристам и водителям оценить друг друга перед поездкой, что повышает доверие между ними.
- Отображение контекстной информации. Для текущего местоположения туриста выводится информация о текущей ситуации. Такой информацией являются погодные условия (облачность, температура, скорость и направление ветра, давление и влажность), отметка текущего местоположения на карте с отметками объектов в ближайшем радиусе, фотографии, сделанные рядом.

В рамках осуществления перехода к концепции открытых данных функциональность интеллектуального туристического гида была расширена для использования в других регионах. В частности, были подключены сервисы, позволяющие осуществлять планирование маршрутов по Санкт-Петербургу, Оулу и Ярославлю. Использование открытых источников информации позволяет осуществлять поиск достопримечательностей, их описаний и фотографий по всему миру.

На рисунке 49 представлены примеры интерфейса пользователя интеллектуального туристического гида для Санкт-Петербурга, отображающие его основную функциональность.

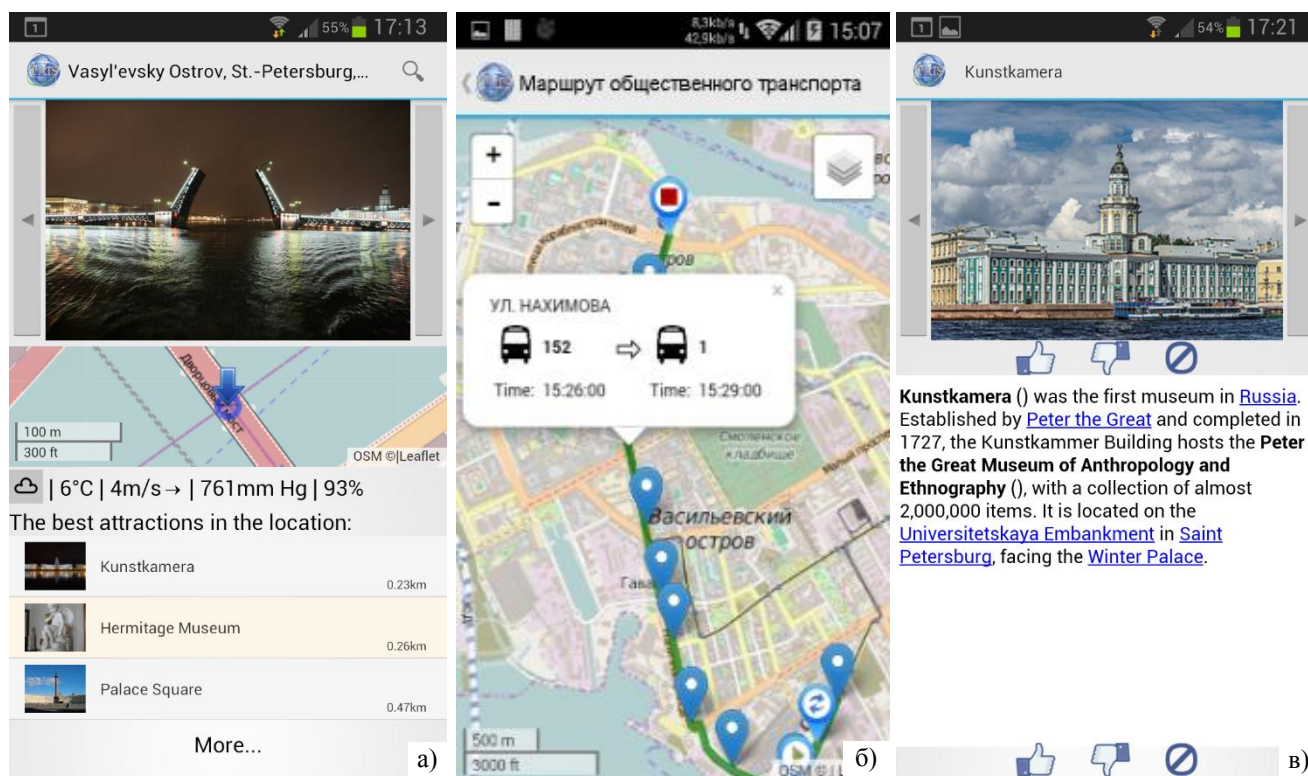


Рисунок 49 — Примеры интерфейса пользователя интеллектуального гида

Рисунок 49 (а) представляет главное окно приложения, на которое выводится текущее местоположение пользователя, контекстная информация (погода и фотографии), а также список из трех достопримечательностей, рекомендованных к посещению. Каждый элемент интерфейса является интерактивным: при нажатии на какой-либо элемент переводит на соответствующий экран: альбом с фотографиями, интерактивную карту, прогноз погоды на ближайшее время и описание достопримечательности.

На рисунке 49 (б) представлен пример планирования мультимодального маршрута. Пользователю отображается полная информация о маршруте, включающая в себя остановки, через которые проедет транспорт, остановки на которых следует совершить пересадку, время пересадки, используемые типы транспортных средств с указанием номеров маршрутов.

На рисунке 49 (в) представлен экран, содержащий информацию об объекте, включающую в себя описание объекта и его фотографии. Описание и фотографии могут быть отдельно оценены пользователем для улучшения качества предоставляемой информации в будущем.

Интеллектуальный туристический гид был также протестирован в других регионах Российской Федерации и мира. На рисунке 50 представлены результаты работы интеллектуального гида в городе Оулу, республики Финляндия (рисунок 50 (а), (б)) и в городе Ярославле, Ярославской области, Российской Федерации (рисунок 50 (в)).

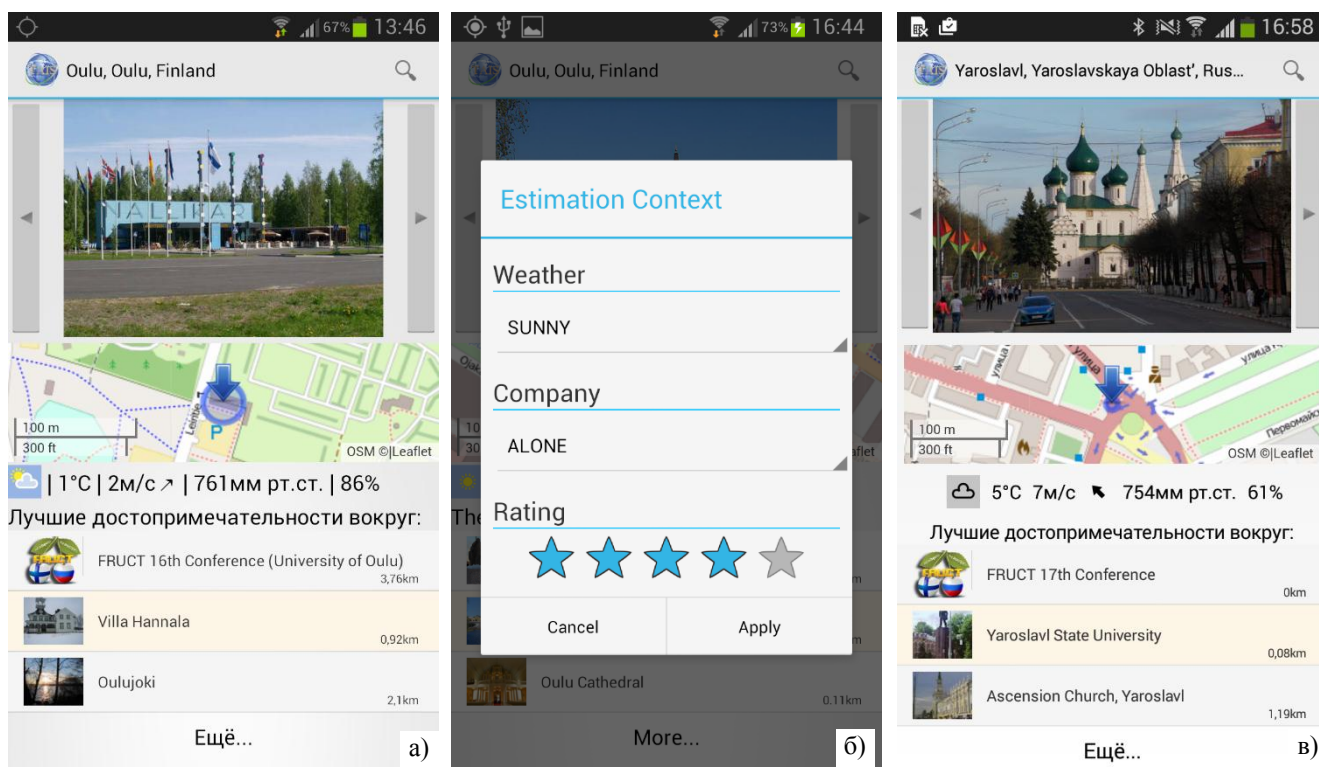


Рисунок 50 — Примеры работы интеллектуального гида в других регионах

В настоящее время интеллектуальный туристический гид, обеспечивающий инфомобильность туристов, разработан только для устройств под управлением операционной системы Android. Последняя версия приложения доступна в магазине приложений Google Play [24] и насчитывает более 1000 установок.

Результаты исследования, связанные с построением онтологической и сценарной модели, а также метод сопоставления онтологий были применены в ЦНИИ РТК в исследованиях, посвященных разработке систем управления робототехническими системами в среде облачных вычислений.

По разработанным сервисам также были получены свидетельства о регистрации программ для ЭВМ «Система обеспечения безопасности личных данных в интеллектуальных пространствах» № 2015619362 от 01.09.2015 и «Сервис рекомендации достопримечательностей для системы обеспечения инфомобильности» № 2015619363 от 01.09.2015.

#### 4.7 Выводы по главе 4

- 1) В главе представлены особенности реализации некоторых сервисов системы, потребовавшие особого внимания при проверке результатов, полученных в предыдущих главах.
- 2) При разработке сервисов, работающих с картографической информацией, таких как сервис планирования маршрутов и сервис поиска объектов по местоположению потребовалась организация собственной ГИС. Требование возникло в силу того,

что сервисы, доступные через сеть Интернет, имеют ограничения на количество запросов, осуществляемых за определенный интервал времени. Для организации собственной ГИС использовались картографические данные, предоставляемые краудсорсинговым проектом OpenStreetMap. Функциональность полученной ГИС включает в себя: хранение, визуализацию и распространение карты; планирование маршрутов с использованием различных видов транспорта; поиск объектов по местоположению.

- 3) Для обеспечения возможности планирования маршрутов из картографической информации, хранящейся в ГИС, была извлечена информация о дорогах, по которым была построена топология, пригодная для работы алгоритмов поиска путей. Из множества классов дорог при поиске пути выбираются классы, соответствующие заданному транспортному средству, что позволяет существенно ускорить процесс поиска.
- 4) Информация о маршрутах общественного транспорта извлекается из свободно распространяемых данных о маршрутах в формате GTFS. Эта информация используется для построения мультиграфа, по которому осуществляется планирование маршрута. При планировании учитывается расписание движения общественного транспорта для минимизации времени ожидания транспортного средства на остановках.
- 5) Реализация метода сопоставления онтологий в соответствующем сервисе осуществляется с помощью библиотек `difflib`, `fuzzywuzzy` и `python-levenshtein`, позволяющих определить степень сходства строк и набора инструментов для работы с естественным языком NLTK, позволяющего определить степень сходства понятий на основе словарной базы WordNet. В случае невозможности сопоставления понятий онтологии формируется микрозадание для исполнения человеком, которое размещается в краудсорсинговой платформе Amazon Mechanical Turk
- 6) Для формирования виртуального приватного интеллектуального пространства используется асимметричный алгоритм шифрования RSA с длиной ключа 2048 бит. Данный алгоритм применяется не только для шифрования передаваемой информации, но и для создания цифровой подписи к ней, с помощью чего можно аутентифицировать отправителей.
- 7) Результаты, полученные в ходе диссертационного исследования, были применены при разработке интеллектуального туристического гида в рамках международного проекта по развитию туризма. Гид позволяет повысить мобильность туриста при



перемещении по региону между достопримечательностями и предоставить ему дополнительные услуги, связанные с поиском информации о них.

## Заключение

В диссертации предложено решение актуальной научно-технической задачи создания сервис-ориентированной системы обеспечения инфомобильности, позволяющей повысить качество обслуживания пользователей как при планировании поездки, так и во время нее. В процессе решения данной задачи были получены результаты, составляющие научную новизну настоящей диссертации, среди которых можно выделить:

- 1) Разработан контекстно-зависимый подход к построению СОИМ, позволяющий обеспечить использование открытых данных о транспорте, открытых информационных транспортных и геоинформационных сервисах в независимых друг от друга сервисах, взаимодействующих между собой для выработки общего решения, учитывающего контекст текущей ситуации вокруг пользователя.
- 2) Разработаны концептуальная, онтологическая и сценарная модели СОИМ, на основе которых была разработана архитектура СОИМ, позволяющая использовать модель «классная доска» для обеспечения асинхронного взаимодействия независимых сервисов, представлением знаний с помощью онтологии для организации взаимодействия сервисов. Сервисы способны описывать свой контекст и изменять поведение в ответ на изменение контекста с целью предоставления услуг релевантных сложившейся ситуации. Данный процесс называется самоконтекстуализацией и является важной характеристикой автономных сервисов.
- 3) Предложена модификация метода сопоставления онтологий, позволяющая использовать технологию краудсорсинга, для привлечения интернет-сообщества к решению задачи автоматизированного сопоставления онтологий в случае невозможности автоматического сопоставления. Для исполнителей формируются микрозадания, позволяющие быстро оценить степень сходства понятий онтологии. Предложенные решения обрабатываются с применением методов математической статистики и используются для корректировки решения, полученного методом автоматического сопоставления онтологий.
- 4) Разработан метод планирования мультимодальных маршрутов, использующий ориентированный взвешенный мультиграф для представления сети маршрутов общественного транспорта в рассматриваемом регионе. Вершинами мультиграфа являются остановки ОТ, а ребрами – проходящие между остановками маршруты с указанием направления движения. Веса ребер задаются динамически и соответствуют времени движения по ребру и стоимости проезда (стоимости билета). В характеристики вершин также входит расписание движения соответствующего

маршрута, что позволяет учитывать его при планировании мультимодального маршрута. В методе также описывается планирование совместных поездок с использованием личного автотранспорта, что позволяет предоставить дополнительный способ передвижения в рассматриваемом регионе и перераспределить транспортный и пассажирский потоки для снижения общей загруженности транспортной инфраструктуры.

- 5) Создан комплекс программных средств для обеспечения инфомобильности в сфере туризма, использующий сервисы, способные к самоконтекстуализации (планирования маршрутов, поиска объектов и информации о них, выработки рекомендаций), а также сервисы для обеспечения конфиденциальности информации пользователей и сопоставления онтологий.

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»

## Определения

В настоящей диссертационной работе применяют следующие термины с соответствующими определениями.

Архитектура «классная доска» — это стратегия решения сложных системных задач с привлечением разнородных источников знаний, взаимодействующих через общее информационное поле.

Инфомобильность — предоставление пользователям доступа с помощью мобильных устройств к мультимодальной динамической персонализированной информации и транспортным сервисам с учетом контекста с целью повышения их мобильности и удобства перемещения как при планировании поездки, так и во время нее.

Информационный ресурс — информация, пригодная для удовлетворения информационных потребностей какого-либо лица и доступная этому лицу.

Информационный транспортный сервис — набор транспортных услуг, предоставляемых с помощью информационных систем при перевозке грузов и пассажиров.

Контекст — любая информация, которая может быть использована для описания ситуации, в которой находится в определенный момент некоторый объект (человек, транспортное средство или сервис).

Онтология — формально представленные на базе концептуализации знания, предполагающие описание множества объектов и понятий, знаний о них и связей между ними.

Самоконтекстуализация — способность сервиса самостоятельно описывать и использовать контекст, а также изменять свое поведение в зависимости от текущей ситуации.

Сервис — механизм для предоставления доступа к одной или многим возможностям, в котором доступ предоставляется с помощью описанного заранее интерфейса и осуществляется в соответствии с ограничениями и политиками, определенными в описании сервиса.

Сервис-ориентированная архитектура — модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных (англ. loose coupling) заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.

## Список обозначений и сокращений

ГИС	—	географическая информационная система
ИП	—	интеллектуальное пространство
ИТС	—	информационный транспортный сервис (интеллектуальная транспортная система)
ОТ	—	общественный транспорт
СОИМ	—	система обеспечения инфомобильности
ТС	—	транспортное средство
AMT	—	Amazon mechanical turk
GTFS	—	General transit feed specification
НIT	—	Human intelligence task
IPC	—	Inter-process communication
LBS	—	Location based service
OSM	—	OpenStreetMap
REST	—	Representation state transfer
SOA	—	Service oriented architecture
SSAP	—	Smart space access protocol

## Литература

- 1 Вучик, В.Р. Транспорт в городах, удобных для жизни [Текст]/ В.Р. Вучик; пер. с англ. Александра Калинина под науч. ред. Михаила Блинкина. — Москва: Территория будущего, 2011. — 574 с.
- 2 Гаврилова, Т. А. Базы знаний интеллектуальных систем: Учебное пособие для вузов [Текст] / Т. А. Гаврилова, В. Ф. Хорошевский. — СПб.: Питер, 2000. — 382 с.
- 3 Городецкий, В. И. Самоорганизация и многоагентные системы I. Модели многоагентной самоорганизации [Текст] / В.И. Городецкий // Известия РАН «Теория и системы управления». — 2012. — №2. — С. 92–120.
- 4 Городецкий, В.И., Прикладные многоагентные системы группового управления [Текст] / В.И. Городецкий, О.В. Карсаев // Искусственный интеллект и принятие решений. — 2009. — №2. — С/ 3-24.
- 5 Довезу!ру [Электронный ресурс]. — Режим доступа: <http://www.dovezu.ru/> (дата обращения: 20.08.2015). (дата обращения: 20.08.2015)
- 6 ЗАО «РИПАС». АСУДД "СПЕКТР" [Электронный ресурс]. — Режим доступа: <http://www.ripas.ru/index.php/produksiya/asudd-spektr> (дата обращения: 20.08.2015)
- 7 Кашевник, А.М. Автоматизированная система совместного использования автотранспорта / А.М. Кашевник, Н.Н. Тесля // Материалы конференции «Информационные технологии в управлении» (ИТУ-2012). — СПб.: ОАО «Концерн «ЦНИИ «Электроприбор». — 2012. — С. 471-479.
- 8 Кашевник, А.М. Архитектура логистической системы поиска попутчиков для водителей [Текст] / А.М. Кашевник, Н.Н. Тесля // Труды СПИИРАН; под ред. Р.М. Юсупова. — СПб.: Наука, 2011. — Вып. 2 (17). — С. 114-150.
- 9 Кашевник, А.М. Разработка интеллектуальных систем на базе платформы Smart-M3 [Текст] / А.М. Кашевник, Д.Ж. Корзун, С.И. Баландин // Учебное пособие для студентов мат. и техн. специальностей вузов. — Петрозаводск: Издательство ПетрГУ, 2013. — 51 с.
- 10 Кашевник, А.М. Разработка рекомендующих систем на основе интеллектуальных пространств: учебное пособие для студентов математических и технических специальностей вузов [Текст] / А. М. Кашевник, Д. Ж. Корзун, С. И. Баландин, А. В. Пономарев. — Петрозаводск.: Изд-во ПетрГУ, 2015. — 71 с.
- 11 Кашевник, А.М. Современные принципы построения систем контекстно-зависимого разграничения доступа к ресурсам интеллектуального пространства [Текст] / А.М. Кашевник, Н.Н. Тесля // Труды 20-ой международной конференции Проблемы

- управления безопасностью сложных систем, — Москва.: Издательский центр РГГУ, 2012. — С. 208-211.
- 12 Левашова, Т.В. Модель контекста в системах интеллектуальной поддержки принятия решений [Текст] / Т.В. Левашова// Поддержка принятия решений: Труды Института системного анализа Российской академии наук / Под ред. А. Б. Петровского. — М.: Издательство ЛКИ, 2008. — Т. 35. — С. 33–42.
- 13 Левенштейн, В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов [Текст] / В.И. Левенштейн// Доклады Академий Наук СССР. — 1965. — 163.4 — С. 845-848.
- 14 Люгер, Д. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем [Текст] / Люгер, Джордж, Ф. — 4-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2003. — 864 с.
- 15 Население: Федеральная служба государственной статистики. [Электронный ресурс] — Режим доступа: [http://www.gks.ru/wps/wcm/connect/rosstat\\_main/rosstat/ru/statistics/population/c](http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/statistics/population/c) (дата обращения: 20.08.2015)
- 16 Пономарев, А.В. Обзор методов учета контекста в системах коллаборативной фильтрации [Текст] / А.В. Пономарев// Труды СПИИРАН. —2013. — Вып. 30. — С. 169-188.
- 17 Портал общественного транспорта г. Санкт-Петербурга [Электронный ресурс]. — Режим доступа: <http://transport.orgp.spb.ru/Portal/transport/main> (дата обращения: 20.08.2015)
- 18 Радченко И.А. Использование открытых данных в научных исследованиях [Текст] / И.А. Радченко // Информационное общество. — 2013. — № 1-2. — С. 93-101.
- 19 Ронжин, Ал. Л., Сравнительный анализ функциональности прототипов интеллектуальных пространств [Текст] / Ал. Л. Ронжин, А. А. Карпов // Труды. СПИИРАН. — 2013. — № 24. — С. 277–290.
- 20 Российская Федерация. Законы. Об информации, информационных технологиях и о защите информации [Текст] : федер. закон : [принят Гос. Думой 8 июля 2006 г. : одобр. Советом Федерации 14 июля 2006 г.] — «Российская газета» № 4131 от 29 июля 2006 г — М.: 2006.
- 21 Смирнов, А.В. Подход к построению распределённой системы интеллектуальной поддержки принятия решений в открытой информационной среде [Текст] / А.В. Смирнов, М.П. Пашкин, Н.Г. Шилов, Т.В. Левашова // Труды СПИИРАН. — СПб.: Наука, 2007. — Вып. 4, т. 1. С. 36 – 49.

- 22 Тесля, Н.Н. Принципы построения интеллектуальных транспортных систем для обеспечения инфомобильности [Текст] / Н.Н. Тесля //Труды СПИИРАН. —2014. — Вып. 6(37). — С. 21-36.
- 23 Транспорт и связь: Федеральная служба государственной статистики. [Электронный ресурс] — Режим доступа: [http://www.gks.ru/wps/wcm/connect/rosstat\\_main/rosstat/ru/statistics/enterprise/transport/](http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/statistics/enterprise/transport/) (дата обращения: 20.08.2015)
- 24 Туристический гид — ТАIS [Электронный ресурс] // Приложения на Google Play. — Режим доступа: <https://play.google.com/store/apps/details?id=ru.nw.spiiras.tais&hl=ru> (дата обращения: 20.08.2015)
- 25 Чечулин, А.А. Особенности построения системы защиты информации в киберфизических системах / А.А. Чечулин, И.В. Котенко, В.А. Десницкий // Методы и технические средства обеспечения безопасности информации. Материалы 23-й научно-технической конференции. — СПб.: Издательство Политехнического университета, 2014. — С.67-69.
- 26 Шилов, Н.Г. Согласование онтологий в групповых рекомендующих системах: полимодельный метод и типовые модели // Труды СПИИРАН. — 2013. — Вып. 24. — С. 313-331.
- 27 Юсупов, Р.М. Концептуальные и научно-методологические основы информатизации [Текст] / Р.М. Юсупов, В.П. Заболотский. — СПб.: Наука, 2009. — 542 с.
- 28 Юсупов, Р.М. От умных приборов к интеллектуальному пространству [Текст] / Р.М. Юсупов, А. Л. Ронжин// Вестник Российской Академии Наук. — 2010. — Том 80, Вып. 1. — С. 45–51.
- 29 Яндекс.Карты [Электронный ресурс]. — Режим доступа: <http://maps.yandex.ru> (дата обращения: 20.08.2015)
- 30 Яндекс.Расписания [Электронный ресурс]. — Режим доступа: <http://rasp.yandex.ru> (дата обращения: 20.08.2015)
- 31 Яндекс.Такси [Электронный ресурс]. Режим доступа: <https://taxi.yandex.ru/taxi/> (дата обращения: 6.02.2013)
- 32 Adomavicius, G. Context-aware recommender systems [Текст] / G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin // AI Magazine. —2011. — vol. 32(3). — pp. 67–80.
- 33 Adomavicius, G. Context-Aware Recommender Systems. [Текст] / G. Adomavicius, A. Tuzhilin // Recommender Systems Handbook. — ed. F. Ricci, L. Rokach, B. Shapira, and P. Kantor. — Springer US, 2011. — Chapter 7. — Pp. 217–256.



- 34 Adomavicius, G. Incorporating contextual information in recommender systems using a multidimensional approach [Текст] / G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin. // ACM Transactions on Information Systems (TOIS). — ACM New York, NY, USA, 2005. — vol. 23, no. 1. — Pp. 103-145.
- 35 Al-Muhtadi, J. Cerberus: a context-aware security scheme for smart spaces [Текст] / J. Al-Muhtadi, A. Ranganathan, R. Campbell, D. Mickunas // Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the 1st IEEE International Conference. — 2003. — Pp. 489-496.
- 36 Alasoud, A. An Effective Ontology Matching Technique [Текст] / A. Alasoud, V. Haarslev, and N. Shiri // 17th International Symposium ISMIS 2008. — 2008. — Pp. 585-590.
- 37 Amazon Mechanical Turk. [Электронный ресурс]. — Режим доступа: <https://www.mturk.com/mturk/welcome> (дата обращения: 20.08.2015)
- 38 Ambrosino, G. Introduction [Текст] / G. Ambrosino, M. Voero, J. D. Nelson, M. Romanazzo // Infomobility Systems and sustainable transport services / G. Ambrosino, M. Voero, J. D. Nelson, M. Romanazzo. — ENEA, 2010. — Chapter 1. — 340 p.
- 39 AnHai, D. Ontology Matching: A Machine Learning Approach [Текст] / D. AnHai, M. Jayant, D. Pedro, and H. Alon // Handbook on Ontologies in Information Systems. — 2004.
- 40 Arikawa, M. Navitime: Supporting Pedestrian Navigation in the Real World [Текст] / M. Arikawa, S. Konomi, K. Ohnishi, // Pervasive Computing, 6(3). — IEEE, 2007. — Pp. 21-29.
- 41 Artemis JU [Электронный ресурс]. — Режим доступа: [http://www.artemis-ju.eu/home\\_page](http://www.artemis-ju.eu/home_page) (дата обращения: 20.08.2015)
- 42 Aumuller, D. Schema and Ontology Matching with COMA++ [Текст] / D. Aumuller, H. Do, S. Massmann, and E. Rahm // Proceedings of the 2005 ACM SIGMOD international conference on Management of data. — 2005. — Pp. 906-908.
- 43 Avego [Электронный ресурс]. — Режим доступа: <http://www.avego.com/> (дата обращения: 20.08.2015)
- 44 Berners-Lee, T. RFC 3986 – Uniform Resource Identifier (URI): Generic Syntax [Электронный ресурс] / T. Berners-Lee, R. Fielding, L. Masinter // Internet Standard. — 2005. — Режим доступа: <http://tools.ietf.org/html/rfc3986> (дата обращения: 20.08.2015)
- 45 Berners-Lee, T. The Semantic Web [Текст] / T. Berners-Lee, J. Hendler O. Lassila // Scientific American. — 2001. — Vol. 284, no. 5. — P. 34–43.
- 46 Beta Features/Nearby Pages [Электронный ресурс] // MediaWiki. — Режим доступа: [https://www.mediawiki.org/wiki/Beta\\_Features/Nearby\\_Pages](https://www.mediawiki.org/wiki/Beta_Features/Nearby_Pages)
- 47 Bird, S. Natural Language Processing with Python [Текст] / S. Bird, E. Loper, E. Klein. — O'Reilly Media, 2009. — 504 p.

- 48 BlaBlaCar – поиск надежных попутчиков [Электронный ресурс]. — Режим доступа: <https://www.blablacar.ru/> (дата обращения: 20.08.2015)
- 49 Boccardo, P. Application of geomatic techniques in infomobility and intelligent transport systems (ITS) [Текст] / P. Boccardo, F. Arneodo, D. Botta // *European Journal of Remote Sensing*. — 2014 — vol. 47, issue 1. — Pp. 95-115.
- 50 Brennan, S. STIS: Smart Travel Planning Across Multiple Modes of Transportation [Текст] / S. Brennan, R. Meier // *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. — 2007. — Pp. 666-671.
- 51 Bui, V. A body sensor platform for concurrent applications [Текст] / V. Bui, R. Verhoeven, J. Lukkien // *Proc. of IEEE Int. Conf. on Consumer Electronics*. — IEEE, 2012. — Pp. 38-42.
- 52 Bui, V. Semantic Interoperability in Body Area Sensor Networks and Applications [Текст] / V. Bui, P. Brandt, H. Liu, T. Basten, J. Lukkien // *9th International Conference on Body Area Networks*. — Great Britain, 2014. — Pp. 210-216.
- 53 Canali, C. A distributed architecture to support infomobility services [Электронный ресурс] / C. Canali, R. Lancellotti // *Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications, (Pisa, Italy), ACM International Conference Proceeding Series*. — ACM, New York, NY, USA, 2006. — Режим доступа: <http://doi.acm.org/10.1145/1190183.1190189>
- 54 CarJungle [Электронный ресурс]. — Режим доступа: <http://www.carjungle.ru/> (дата обращения: 20.08.2015)
- 55 Carminati, B. A Semantic Web Based Framework for Social Network Access Control [Текст] / B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, B. Thuraisingham // *Proceedings of the 14th ACM symp. on Access control models and technologies*. — 2009. — Pp. 177-186.
- 56 Carminati, B. Semantic web-based social network access control [Текст] / B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, B. Thuraisingham // *Comp. & Security*. — 2011. — vol. 30, issues 2–3. — Pp. 108–115.
- 57 Carpool [Электронный ресурс] // Wikipedia. — Режим доступа: <http://en.wikipedia.org/wiki/Carpool> (дата обращения: 20.08.2015).
- 58 Castano, S. H-Match: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems [Текст] / S. Castano, A. Ferrara, and S. Montanelli // *Proc. of the 1st VLDB Int. Workshop on Semantic Web and Databases*. — 2003.
- 59 Chandrasekaran, B. What Are Ontologies, and Why Do We Need Them? [Текст] / B. Chandrasekaran, J.R. Josephson V. R. Benjamins // *IEEE Intelligent Systems*. — 1999. — Pp. 20–25.

- 60 CHIRON project: Cyclic and person-centric Health management, [Электронный ресурс]. — Режим доступа: <http://www.chiron-project.eu/> (дата обращения: 20.08.2015).
- 61 Cooper, D. RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile [Электронный ресурс] / D. Cooper, S. Santesson, S. Farrell, S. Voeyen, R. Housley, W. Polk // Proposed Standard — 2008. — Режим доступа: <http://tools.ietf.org/html/rfc5280> (дата обращения: 20.08.2015).
- 62 Cruz, I. Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods [Текст] / I. Cruz, F. Antonelli, C. Stroe // The Fourth International Workshop on Ontology Matching, Washington DC. — 2009. — Pp. 1-12.
- 63 Definition of crowdsourcing [Электронный ресурс] // Merriam-Webster. — Режим доступа: <http://www.merriam-webster.com/dictionary/crowdsourcing> (дата обращения: 20.08.2015)
- 64 Development of cross-border e-tourism framework for the programme region (Smart e-Tourism) [Электронный ресурс]. — Режим доступа: [http://www.kareliaenpi.eu/en/themes/tourism-cooperation/projects/242\\_](http://www.kareliaenpi.eu/en/themes/tourism-cooperation/projects/242_)(дата обращения: 20.08.2015)
- 65 Dewan, K.K. Carpooling: A Step To Reduce Congestion (A Case Study of Delhi) [Текст] / K. K.Dewan, I. Ahmad // International MultiConference of Engineers & Computer Scientists. — Newswood Limited, 2006. — Pp. 408-413.
- 66 Dey, A.K. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications [Текст] / Dey, A.K., Salber, D. Abowd, G.D. // A special issue on Context-Aware Computing, Human-Computer Interaction (HCI) Journal. — Taylor & Francis, 2001. — Vol. 16 (2-4). — Pp. 97-166.
- 67 DiffLib — Helpers for computing deltas [Электронный ресурс] // Python 2.7.10 documentation. — Режим доступа: <https://docs.python.org/2/library/difflib.html#module-difflib> (дата обращения: 20.08.2015)
- 68 Directive 2010/40/EU of the European parliament and of the council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport [Электронный ресурс] — Режим доступа: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF> (дата обращения: 20.08.2015)
- 69 Doan, A. Learning to map between ontologies on the semantic web [Текст] / A. Doan, J. Madhavan, P. Domingos, and A. Halevy // Proceedings of the 11th international conference on World Wide Web. — 2002. — Pp. 662-673.
- 70 Eastlake, D. RFC 3174 - US Secure Hash Algorithm 1 (SHA1) [Электронный ресурс] / D. Eastlake, P. Jones // Informational. — 2001. — Режим доступа: <https://tools.ietf.org/html/rfc3174> (дата обращения: 20.08.2015)

- 71 eRideShare.com [Электронный ресурс]. — Режим доступа: <http://erideshare.com/> (дата обращения: 20.08.2015).
- 72 Erl, T. Service-Oriented Architecture Principles [Электронный ресурс] / T. Erl. // ServiceOrientation. — Режим доступа: <http://serviceorientation.com/serviceorientation/index> (дата обращения: 20.08.2015)
- 73 Erl, T. SOA Principles of Service Design [Текст] / T. Erl. — Prentice Hall/PearsonPTR, 2007. — 573 p.
- 74 Fabini, J. Terminal-Centric Location Services for the IP Multimedia Subsystem [Текст] / J. Fabini, M. Happenhofer, R. Pailer // Proceedings of Vehicular Technology Conference. — IEEE, 2006. — Vol. 2. — Pp. 881-885.
- 75 Fensel, D. Relating ontology languages and web standards [Текст] / D. Fensel // Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik (Modellierung 2000); eds. by J. Ebert, U. Frank. — Koblenz: Foelbach-Verlag, 2000. — P. 111–128.
- 76 Foursquare [Электронный ресурс]. — Режим доступа: <https://ru.foursquare.com/> (дата обращения: 20.08.2015)
- 77 Fuzzy String Matching in Python [Электронный ресурс] // GitHub. — Режим доступа: <https://github.com/seatgeek/fuzzywuzzy> (дата обращения: 20.08.2015)
- 78 García, C.R. Architecture of a Framework for Providing Information Services for Public Transport [Текст] / C.R. García, R. Pérez, A. Lorenzo, A. Quesada-Arencibia, F. Alayón, G. Padrón // Sensors (Switzerland). — May 2012. — vol. 12, issue 5. — Pp. 5290-5309.
- 79 Garnaat, M. boto: A Python interface to Amazon Web Services [Электронный ресурс] / M. Garnaat. — 2010. — Режим доступа: <https://boto.readthedocs.org/en/latest/> (дата обращения: 20.08.2015)
- 80 General transit feed specification (GTFS) [Электронный ресурс] // Google Developers. — Режим доступа: <https://developers.google.com/transit/gtfs/> (дата обращения: 20.08.2015)
- 81 GeoNames [Электронный ресурс]. — Режим доступа: <http://www.geonames.org/> (дата обращения: 20.08.2015)
- 82 Gett [Электронный ресурс]. — Режим доступа: <http://gett.com/ru/> (дата обращения: 20.08.2015)
- 83 GISmeteo прогноз погоды [Электронный ресурс]. — Режим доступа: <https://www.gismeteo.ru/> (дата обращения: 20.08.2015)
- 84 Giunchiglia, F. Computing minimal mappings between lightweight ontologies [Текст] / F. Giunchiglia, V. Maltese, A. Autayeu // International Journal on Digital Libraries — Springer-Verlag, 2012. — Vol. 12, issue 4. — Pp. 179-193.

- 85 Google Maps [Электронный ресурс]. — Режим доступа: <http://maps.google.com>\_(дата обращения: 20.08.2015)
- 86 GraphHopper. [Электронный ресурс]. — Режим доступа: <http://graphhopper.com/> (дата обращения: 20.08.2015)
- 87 Guha S. Rock: A robust clustering algorithm for categorical attributes [Текст] / S. Guha, R. Rastogi, K. Shim // *Information Systems*. — Elsevier, 2000. — Vol. 25, issue 5. — Pp. 512–521.
- 88 Hanif, M.S. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size [Текст] / M. S. Hanif, M. Aono // *Journal of Web Semantics*. — Elsevier, 2009. — Vol. 7, no. 4. — Pp. 344-356.
- 89 Happel, H.-J. Applications of Ontologies in Software Engineering [Электронный ресурс] / H.-J. Happel, S. Seedorf // *Proc. Second Int'l Workshop Semantic Web Enabled Software Eng.* — 2006. Режим доступа: [https://km.aifb.kit.edu/ws/swese2006/final/happel\\_full.pdf](https://km.aifb.kit.edu/ws/swese2006/final/happel_full.pdf) (дата обращения: 20.08.2015)
- 90 He, Z. Semantics-based Access Control Approach for Web Service [Текст] / Z. He, L. Wu, H. Li, H. Lai, Z. Hong // *Journal of Comp.* — 2011. — vol. 6, no. 6. — Pp. 1152-1161.
- 91 Here [Электронный ресурс]. — Режим доступа: <http://here.com>\_(дата обращения: 20.08.2015)
- 92 Honkola, J. Smart-M3 Information Sharing Platform [Текст] / J. Honkola, H. Laine, R. Brown, O. Tyrkko // *Proceedings of the IEEE Symp. Computers and Communications (ISCC'10)*. — IEEE Comp. Soc., 2010. — Pp. 1041-1046.
- 93 Hovy, E. Combining and standardizing largescale, practical ontologies for machine translation and other uses [Текст] / E. Hovy // *The First International Conference on Language Resources and Evaluation (LREC)*. — 1998. — Pp. 535–542.
- 94 Hu, W. GMO: A Graph Matching for Ontologies [Текст] / W. Hu, N. Jian, Y. Qu, Y. Wang // *K-CAP Workshop on Integrating Ontologies*. — 2005. — Pp. 43-50.
- 95 Hu, W. Matching large ontologies: A divide-and-conquer approach [Текст] / W. Hu, Y. Qu, G. Cheng // *Data and Knowledge Engineering*. — 2008. — Vol. 67, no. 1. — Pp. 140-160.
- 96 Huang, D. MapWeb: A location-based converged communications platform [Текст] / D. Huang, F. Liu, X. Shi, G. Yang, L. Zheng, Z. Zhou // *Bell Labs Technical Journal*. — Wiley Periodicals, 2006. — Vol. 11, issue 1. — Pp. 159-171.
- 97 IOE project: Internet of Energy [Электронный ресурс]. — Режим доступа: <http://www.artemis-ioe.eu>\_(дата обращения: 20.08.2015).

- 98 Jean-Mary, Y.R. Ontology matching with semantic verification [Текст] / Y. R. Jean-Mary, E.P. Shironoshita, and M. R. Kabuka // Journal of Web Semantics. — Elsevier, 2009. — Vol. 7, no. 3. — Pp. 235-251.
- 99 Jennings, C. Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks (IETF Request For Comment: 3325) [Электронный ресурс] / C. Jennings, J. Peterson, M. Watson. — 2002. — Режим доступа: <http://www.ietf.org/rfc/rfc3325.txt> (дата обращения: 20.08.2015)
- 100 Josuttis, N.M. SOA in practice: the art of distributed system design [Текст]/ N.M. Josuttis. — Sebastopol, California: O'Reilly Media, Inc., 2007. — 344 p.
- 101 Karelia ENPI CBC [Электронный ресурс] // Карелия. Туристский портал. — Режим доступа: [http://www.ticrk.ru/ru/section\\_33197.html](http://www.ticrk.ru/ru/section_33197.html) (дата обращения: 20.08.2015).
- 102 Klein, M. Ontology Versioning and Change Detection on the We [Текст] / M. Klein, W. Kiryakov, D. Ognyanov, D. Fensel // 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02). — Spain, 2002. — Pp. 197-212.
- 103 Koljonen, T. ARTEMIS and the rest of the world [Текст] / T. Koljonen // ARTEMIS Magazine. — 2009, No. 5. — Pp. 30–31.
- 104 Korzun, D. Proactive personalized mobile multiblogging service on SmartM3 [Текст] / D. Korzun, I. Galov, S. Balandin// Journal of Computing and Information Technology. — 2012. — vol. 20, No. 3. — Pp. 175-182.
- 105 Kuhn, D. R. Adding Attributes to Role-Based Access Control [Текст] / D. R. Kuhn, E. J. Coyne, T. R. Weil // IEEE Computer. — 2010. — vol. 43, no. 6. — Pp. 79-81.
- 106 Lambrix, P. SAMBO – a system for aligning and merging biomedical ontologies [Текст] / P. Lambrix, H. Tan // Journal of Web Semantics. — 2006. — Vol. 4, no. 1. — Pp. 196-206.
- 107 Leaflet. An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps [Электронный ресурс]. — Режим доступа: <http://leafletjs.com/> (дата обращения: 20.08.2015)
- 108 Levenshtein Python C extension module [Электронный ресурс] // GitHub. — Режим доступа: <https://github.com/ztane/python-Levenshtein/> (дата обращения: 20.08.2015)
- 109 Li, J. Rimom: A dynamic multistrategy ontology alignment framework [Текст] / J. Li, J. Tang, Y. Li, Q. Luo // IEEE Transactions on Knowledge and Data Engineering. — IEEE, 2009. — Vol. 21, no. 8. — Pp. 1218-1232.
- 110 Liuha, P. Smart Objects for Intelligent Applications [Текст] / Petri Liuha, Antti Lappeteläinen, Juha-Pekka Soininen // ARTEMIS magazine. — 2009. — №. 5. — pp. 27-29.
- 111 Live ADS-B Flight Data Feed [Электронный ресурс] // FlightAware. — Режим доступа: <http://ru.flightaware.com/commercial/data/adsb#sample-json> (дата обращения: 20.08.2015)

- 112 Марник. [Электронный ресурс]. — Режим доступа: <http://marnik.org/> (дата обращения: 20.08.2015).
- 113 Mazzucchelli, L. Extensible Interfaces for Mobile Devices in an Advanced Platform for Infomobility Services [Текст] / L. Mazzucchelli, M. Pace // Proceedings of the working conference on Advanced visual interfaces. — ACM New York, NY, USA 2004. — Pp. 450-453.
- 114 McGuinness, D.L. An Environment for Merging and Testing Large Ontologies [Текст] / D.L. McGuinness, R. Fikes, J. Rice, and S. Wilder // Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000). —USA, 2000. — Режим доступа: <http://www.ksl.stanford.edu/software/chimaera/> (дата обращения: 20.08.2015).
- 115 Melnik, S. Similarity flooding: a versatile graph matching algorithm and its application to schema matching [Текст] / S. Melnik, H. Garcia-Molina, E. Rahm // Proceedings. 18th International Conference on Data Engineering. — USA, 2002. — Pp. 117-128.
- 116 Mitra, P. Graph-Oriented Model for Articulation of Ontology Interdependencies [Текст] / P. Mitra, M. Kersten, G. Wiederhold // Lecture Notes In Computer Science — Springer-Verlag, 2000. — Vol. 1777. — Pp. 86-100.
- 117 Mitra, P. Semi-automatic Integration of Knowledge Sources [Текст] / P. Mitra, G. Wiederhold, J. Jannink // 2nd International Conference on Information Fusion, Sunnyvale. — 1999.
- 118 Mohammad, A. Ontology-Based Access Control Model for Semantic Web service [Текст] / A. Mohammad, G. Kanaan, T. Khdour, S. Bani-Ahmad // Journal of Information And Computing Science. —2011. — vol. 6, no. 3. — Pp. 177-194.
- 119 Moraitis, P. An Agent-Based System for Infomobility Services [Текст]/ P. Moraitis, E. Petraki, N.I. Spanoudakis // 3rd European Workshop on Multi-Agent Systems (EUMAS2005). — 2005. — Pp. 224-235.
- 120 Moraitis, P. Providing Advanced, Personalised Infomobility Services Using Agent Technology [Текст] / P. Moraitis, E. Petraki, N.I. Spanoudakis // Applications and Innovations in Intelligent Systems XI. — 2004. — pp 35-48.
- 121 Moskowitz, R. RFC 5201: Host Identity Protocol [Электронный ресурс] / R. Moskowitz, P. Nikander, P. Jokela, T. Henderson. Experimental. — 2008. — Режим доступа: <http://tools.ietf.org/html/rfc5201> (дата обращения: 20.08.2015).
- 122 Mosmondor, M. Bringing Location Based Services to IP Multimedia Subsystem [Текст] / M. Mosmondor, L. Skorin-Kapov, M. Kovacic // Proceedings of Mediterranean Electrotechnical Conference MELECON 2006. — IEEE, 2006. — Pp. 746-749.
- 123 Nagy, M. Towards an automatic semantic data integration: Multi-agent framework approach [Текст] / M. Nagy, M. Vargas-Vera // Semantic Web. — 2010. — Ch. 7. — Pp. 107-134.

- 124 Natvig, M.K. National multimodal travel information – a strategy based on stakeholder involvement and intelligent transportation system architecture [Текст] / M.K. Natvig, H. Westerheim, // Intelligent Transportation Systems. — ИЕТ, 2007. — Vol. 1, issue 2. — Pp. 102–109.
- 125 Noy, N. Anchor-PROMPT: Using Non-Local Context for Semantic Matching [Текст] / N. Noy, M. Musen // Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence. — USA, 2001. — Pp. 63-70.
- 126 Noy, N. SMART: Automated Support for Ontology Merging and Alignment [Текст] / N. Noy, M. Musen // 12th Workshop on Knowledge Acquisition, Modeling, and Management. — 1999. — Pp. 1-24.
- 127 Oliver, I. Clouds, Spaces and Information Sharing - A Future for the Semantic Web [Электронный ресурс] / I. Oliver // 5th Conf. of Open Innovations Framework Program FRUCT. — Режим доступа: [http://www.fruct.org/sites/default/files/files/seminar5/s5\\_Fruct\\_IanOliver\\_29April2009.pdf](http://www.fruct.org/sites/default/files/files/seminar5/s5_Fruct_IanOliver_29April2009.pdf). (дата обращения: 20.08.2015).
- 128 Open Mobile Alliance. Location Protocol Specification (v. 3.0). [Электронный ресурс] — 2007. — Режим доступа: <http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html> (дата обращения: 20.08.2015).
- 129 OpenStreetMap. [Электронный ресурс]. — Режим доступа: <http://www.openstreetmap.org/> (дата обращения: 20.08.2015).
- 130 OpenTraffic [Электронный ресурс] // Doroga.tv. — Режим доступа: <http://cloud.doroga.tv/opentraffic/> (дата обращения: 20.08.2015).
- 131 Osm2pgsql/schema [Электронный ресурс] // OpenStreetMap Wiki. — Режим доступа: <http://wiki.openstreetmap.org/wiki/Osm2pgsql/schema> (дата обращения: 20.08.2015).
- 132 Paganelli, F. An Evaluation of Context-Aware Infomobility Systems [Текст] / F. Paganelli, D. Giuli // In D. Stojanovic (Ed.), Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications. Hershey, PA: Information Science Reference, 2009 — Pp. 338-361.
- 133 pgRouting. [Электронный ресурс]. — Режим доступа: <http://pgrouting.org/> (дата обращения: 20.08.2015).
- 134 PickupPal [Электронный ресурс]. — Режим доступа: <http://www.pickupal.com/> (дата обращения: 20.08.2015).
- 135 PostgreSQL. [Электронный ресурс]. — Режим доступа: <http://www.postgresql.org/> (дата обращения: 20.08.2015).



- 136 PostGIS. [Электронный ресурс]. — Режим доступа: <http://postgis.refrations.net/> (дата обращения: 20.08.2015).
- 137 Principles of Service Oriented Design [Электронный ресурс] // Microsoft Software Development Network. — Режим доступа: <https://msdn.microsoft.com/en-us/library/bb972954.aspx> (дата обращения: 20.08.2015)
- 138 PTV Partner. Optima [Электронный ресурс]. — Режим доступа: [http://www.ptv-vision.ru/produkty/optima\\_](http://www.ptv-vision.ru/produkty/optima_) (дата обращения: 20.08.2015)
- 139 Pure-Python RSA implementation [Электронный ресурс] // Python Package Index — Режим доступа: <https://pypi.python.org/pypi/rsa> (дата обращения: 20.08.2015)
- 140 Raz, D. Fast and efficient context-aware services [Текст] / D. Raz, A. T. Juhola, J. Serrat-Fernandez, A. Galis. — John Wiley & Sons Ltd, 2006. — 222 p.
- 141 Rehr, K. Assisting Multimodal Travelers: Design and Prototypical Implementation of a Personal Travel Companion [Текст] / K. Rehr, S. Bruntsch, H.J. Mentz // IEEE Transactions on Intelligent Transportation Systems, 8(1). — IEEE, 2007. — Pp. 31-42.
- 142 Resource Description Framework (RDF) [Электронный ресурс] // RDF Working Group. — 2014. — Режим доступа: <http://www.w3.org/RDF/> (дата обращения: 20.08.2015)
- 143 Rideshare 511 [Электронный ресурс]. — Режим доступа: <http://rideshare.511.org/> (дата обращения: 20.08.2015).
- 144 Rideshare History & Statistics. MIT "Real-Time" Rideshare Research. [Электронный ресурс]. — Режим доступа: <http://ridesharechoices.scripts.mit.edu/home/histstats/> (дата обращения: 20.08.2015).
- 145 RideshareOnline [Электронный ресурс]. — Режим доступа: <http://www.rideshareonline.com/> (дата обращения: 20.08.2015).
- 146 Ritze, D. A pattern-based ontology matching approach for detecting complex correspondences [Текст] / D. Ritze, C. Meilicke, O. Šváb-Zamazal, H. Stuckenschmidt // The Fourth International Workshop on Ontology Matching, Washington DC. — 2009. — Pp. 1-12.
- 147 Rónai, M. A. Integrated Design for Context, Network and Policy Management [Электронный ресурс] / M. A. Rónai et al. // Sixth Framework Program Priority IST-2004-2.4.5 Mobile and Wireless Systems beyond 3G Project 027662 “Ambient Networks Phase 2”. — 2006. — 197 p. — Режим доступа: [http://www.ronai.hu/doc/deliverables/AmbientNetworks-D10-D.1\\_PU.pdf](http://www.ronai.hu/doc/deliverables/AmbientNetworks-D10-D.1_PU.pdf) (дата обращения: 20.08.2015).
- 148 RSA Laboratories. PKCS #1 v2.2: RSA Cryptography Standard [Электронный ресурс] / RSA Laboratories. — 2012. — Режим доступа: <https://www.emc.com/collateral/white-papers/h11300-pkcs-1v2-2-rsa-cryptography-standard-wp.pdf> (дата обращения: 20.08.2015).

- 149 RTSoft. Зональные центры управления дорожным движением «СТАРТ-КВИН» [Электронный ресурс]. — Режим доступа: <http://www.rtsoft.ru/projects/detail.php?ID=61> (дата обращения: 20.08.2015).
- 150 RU:Key:highway [Электронный ресурс] // OpenStreetMap Wiki. — Режим доступа: <http://wiki.openstreetmap.org/wiki/RU:Key:highway> (дата обращения: 20.08.2015).
- 151 Saleemi, M.M. A Framework for Context-aware Applications for Smart spaces [Текст] / M. Mohsin Saleemi, Natalia Diaz Rodriguez, Johan Lilius and Ivan Porres // Smart spaces and Next Generation Wired/Wireless Networking. 11th Int. Conf., NEW2AN 2011, and 4th Conf., ruSMART 2011. — St. Petersburg, Russia, 2011. — Pp. 14-25.
- 152 SCATS: The benchmark in urban traffic control [Электронный ресурс]. — Режим доступа: <http://www.scats.com.au/> (дата обращения: 20.08.2015).
- 153 Semantic Web: [Электронный ресурс]. Режим доступа: [http://semanticweb.org/wiki/Main\\_Page](http://semanticweb.org/wiki/Main_Page) (дата обращения: 20.08.2015).
- 154 Serafini, L. An algorithm for matching contextualized schemas via SAT [Текст] / L. Serafini, P. Bouquet, B. Magnini, S. Zanobini // Technical report, DIT University of Trento. — Italy, 2003. — 8 p.
- 155 Slippy Map. OpenStreetMap Wiki. [Электронный ресурс]. — Режим доступа: [http://wiki.openstreetmap.org/wiki/Slippy\\_Map](http://wiki.openstreetmap.org/wiki/Slippy_Map) (дата обращения: 20.08.2015).
- 156 Smart-M3 at Sourceforge. [Электронный ресурс]. — Режим доступа: <http://sourceforge.net/projects/smart-m3> (дата обращения: 20.08.2015).
- 157 Smirnov, A. Infomobility for “car-driver” systems: Reference model and case study [Текст] / A. Smirnov, A. Kashevnik, N. Shilov // IFIP Advances in Information and Communication Technology. — 2014. — vol. 434. — Pp. 739-748.
- 158 Smirnov, A. On-the-Fly Ontology Matching in Smart Spaces: A Multi-model Approach [Текст] / A. Smirnov, A. Kashevnik, N. Shilov, S. Balandin, I. Oliver, S. Boldyrev // Smart Spaces and Next Generation Wired/Wireless Networking Lecture Notes in Computer Science. — Springer Berlin Heidelberg, 2010. — Volume 6294. — Pp 72-83.
- 159 Smirnov, A. OpenStreetMap-Based Dynamic Ridesharing Service [Текст] / A. Smirnov, N. Shilov, A. Kashevnik, N. Teslya // Lecture Notes in Geoinformation and Cartography. — Springer Berlin Heidelberg, 2014. — Pp. 119-134.
- 160 Smirnov, A. Smart Space-based Ridesharing Service in e-Tourism Application for Karelia Region Accessibility Ontology-based Approach and Implementation [Текст] / A. Smirnov, N. Shilov, A. Kashevnik, N. Teslya, S. Laizane // Proceedings of the 8th International Joint Conference on Software Technologies. — SciTePress, 2013. — Pp. 591-598.

- 161 Smirnov, A. Smart Space-Based Tourist Recommendation System: Application for Mobile Devices [Текст] / A. Smirnov, A. Kashevnik, A. Ponomarev, N. Teslya, M. Shchekotov, S.I. Balandin // Lecture Notes in Computer Science. — Springer International Publishing, 2014. — Volume 8638. — Pp 40–51.
- 162 SPARQL Protocol for RDF [Электронный ресурс] // W3C Recommendation. — 2008. — Режим доступа: <http://www.w3.org/TR/rdf-sparql-protocol/> (дата обращения: 20.08.2015).
- 163 SPARQL Query Language for RDF [Электронный ресурс] // W3C Recommendation. — 2008. — Режим доступа: <http://www.w3.org/TR/rdf-sparql-query/> (дата обращения: 20.08.2015).
- 164 St Petersburg Guide Monument [Электронный ресурс]. // Приложения на Google Play. — Режим доступа: <https://play.google.com/store/apps/details?id=com.msinnovations.monumenttracker.stptracker> (дата обращения: 20.08.2015).
- 165 Stoilos, G. A string metric for ontology alignment / G. Stoilos, G. Stamou, and S. Kollias // Lecture Notes in Computer Science. — Springer-Verlag, 2005. — Vol. 3729. — Pp. 624-637.
- 166 Techno-трафик. АСУДД Магистраль [Электронный ресурс]. — Режим доступа: [http://www.technotraffic.ru/solutions/asudd-magistral/asudd-magistral\\_92.html](http://www.technotraffic.ru/solutions/asudd-magistral/asudd-magistral_92.html) (дата обращения: 20.08.2015).
- 167 Teslya, N. Ontology for Resource Self-Organisation in Cyber-Physical-Social Systems [Текст] / N. Teslya, A. Smirnov, T. Levashova, N. Shilov // Communications in Computer and Information Science. — Springer, 2014. — Vol. 468. — Pp. 184-195
- 168 The World Bank group. Motor vehicles (per 1,000 people) [Электронный ресурс] — Режим доступа: <http://data.worldbank.org/indicator/IS.VEH.NVEH.P3> (дата обращения: 14.12.2014)
- 169 TomTom Live Traffic [Электронный ресурс]. — Режим доступа: <http://liveltraffic.tomtom.com/> (дата обращения: 20.08.2015).
- 170 TripAdvisor [Электронный ресурс]. — Режим доступа: <http://www.tripadvisor.ru/> (дата обращения: 20.08.2015).
- 171 Uber [Электронный ресурс]. — Режим доступа: <https://www.uber.com/ru/> (дата обращения: 20.08.2015).
- 172 Umlauft, M. LoL@, a Mobile Tourist Guide for UMTS / M. Umlauft, , G. Pospischil, G. Niklfeld, E. Michlmayr // Information Technology & Tourism. — 2003. — Vol. 5, issue 3. — Pp. 151-164.
- 173 Valipour, M.H. A brief survey of software architecture concepts and service oriented architecture [Текст]/ M. H. Valipour, B. AmirZafari, Kh. N. Maleki, N. Daneshpour // 2009 2nd IEEE International Conference on Computer Science and Information Technology. — IEEE, 2009. — pp. 34–38.

- 174 Verma, S. Comparative analysis of Role Base and Attribute Base Access Control Model in Semantic Web [Текст] / S. Verma, M. Singh, S. Kumar // International Journal of Comput. Applicat. — 2012. — vol. 46, no.18. — Pp. 1-6.
- 175 Wikipedia Nearby [Электронный ресурс]// Wikipedia. — Режим доступа: <https://en.wikipedia.org/wiki/Special:Nearby>\_(дата обращения: 20.08.2015).
- 176 Winkler, W.E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage [Текст] / W.E. Winkler // Proceedings of the Section on Survey Research Methods (American Statistical Association). — 1990. — Pp. 354–359.
- 177 World Weather Online [Электронный ресурс]. — Режим доступа: <http://www.worldweatheronline.com/>\_(дата обращения: 20.08.2015).
- 178 Xiang, Q.J. Framework design of highway traveller information system of Jiangsu Province of China / Q.J. Xiang, Y.F. Ma, J. Lu, J.P. Xie, H.Y. Sha // Intelligent Transportation Systems. — IET, 2007. — Vol.1, issue 2. — Pp. 110-116.
- 179 Yudenok, K. Distributed Service Environment (Smart Spaces) Security Model Development [Текст] / K. Yudenok, K. Krinkin // Proceedings of 12th Conference of Open Innovations Association FRUCT. —St. Petersburg: State University of Aerospace Instrumentation, 2012. — Pp. 210–211.
- 180 Zimride [Электронный ресурс]. — Режим доступа: <http://www.zimride.com/>\_(дата обращения: 20.08.2015).

## Приложение 1. Акты внедрения

---

**УТВЕРЖДАЮ**

Заместитель директора ЦНИИ РТК

  
А.В. Иванов  
«20» 07 2015 г.



### **АКТ О ВНЕДРЕНИИ**

результатов диссертационной работы  
на соискание ученой степени  
кандидата технических наук  
**Тесли Николая Николаевича**

Настоящим Актом удостоверяется, что результаты диссертационной работы Тесли Н.Н., выполненной на тему «Разработка методов и моделей построения сервис-ориентированной системы обеспечения инфомобильности», использованы ЦНИИ РТК в исследованиях, посвященных разработке систем управления робототехническими системами в среде облачных вычислений.

В частности, использованы следующие результаты диссертационной работы:

1. Онтологическая и сценарная модели обеспечения инфомобильности группировки роботов, отличающиеся выделением составляющих контекста (виртуальной и физической) по их источникам и поддержкой самоконтекстуализации сервисов.
2. Модификация метода актуализации онтологий, обеспечивающего семантическую интероперабельность сервисов в системе управления, отличающаяся применением сетцентрического краудсорсинга.

Использование моделей, представленных в диссертационном исследовании, позволяет обеспечить самоконтекстуализацию сервисов, входящих в состав автоматизированной системы управления. Данное свойство было применено в системе управления группой роботов, в которой каждый робот представлен автономным агентом, взаимодействующим с

---


другими роботами с использованием облачных сервисов для принятия решения с учетом контекста выполнения операций.

Описание контекста для каждого робота осуществляется с помощью онтологии, описывающей проблемную область задачи, выполняемой роботом. Применение предложенного в диссертационной работе метода модификации и сопоставления онтологий позволило повысить автономность функционирования робота в автоматическом режиме при проведении исследований в рамках космического эксперимента «Контур-2».


заместитель директора, к.т.н.

  
\_\_\_\_\_ А.С. Кондратьев

начальник лаборатории, к.т.н.

  
\_\_\_\_\_ А.В. Силенко

старший научный сотрудник, к.т.н.

  
\_\_\_\_\_ В.А. Мулюха

ученый секретарь, к.э.н.

  
\_\_\_\_\_ М.М. Буркина



**Российская Федерация  
Республика Карелия  
Государственное бюджетное учреждение  
«Информационный туристский центр Республики Карелия»**

185035 Республика Карелия г. Петрозаводск, ул. Куйбышева 5, тел/факс (814-2) 76-48-35, ИНН/КПП 1001044274/100101001  
УФК по Республике Карелия (Государственное бюджетное учреждение Республики Карелия «Информационный туристский центр Республики Карелия» л/с 21066Х80270) счет 40601810800001000001 ГРКЦ НБ Респ. Карелии Банка России г. Петрозаводск. БИК 048602001

**АКТ**

об использовании результатов  
кандидатской диссертационной работы

**Тесли Николая Николаевича**

Настоящий акт составлен в том, что результаты диссертационной работы **«Разработка методов и моделей построения сервис-ориентированной системы обеспечения информобильности»** использованы при разработке интеллектуального гида для туристов и сервиса совместного использования автотранспорта в рамках проекта КА322 «Smart e-Tourism» финансируемого программой «Карелия» (совместная программа Европейского Союза, России и Финляндии).

Использованы следующие результаты работы:

1. Подход к построению сервис-ориентированной системы обеспечения информобильности.
2. Сценарная, концептуальная и онтологическая модели для организации работы сервисов, предоставляющих информацию о возможном способе перемещения и сервисов, предоставляющих информацию о достопримечательностях региона.
3. Модификация метода сопоставления онтологий, использующая привлечение экспертов при невозможности автоматического сопоставления онтологий
4. Метод планирования мультимодальных маршрутов с учетом расписания движения общественного транспорта и возможностью совместных поездок с использованием личного автотранспорта

Использование приведенных результатов позволило создать информационные системы, позволяющие предоставить широкий спектр информационно-транспортных услуг в сфере туризма на территории Республики Карелия. На основе приведенных результатов были разработаны:

1. Интеллектуальный гид для туристов, объединяющий в едином приложении сервисы для поиска информации о достопримечательностях Республики Карелия и вырабатывающий рекомендации о том, какие достопримечательности и другие объекты интереса можно посетить с учетом предпочтений туриста и текущей ситуации, например, погоды и рабочих часов. Дополнительно предоставляются планы маршрутов до рекомендуемых объектов с использованием видов транспорта, доступных в Республике Карелия;
2. Сервис совместного использования автотранспорта, позволяющий представить альтернативный способ перемещения и позволить туристам найти водителей, желающих разделить поездку с туристами, тем самым снижая общие затраты на поездку.

**Директор**



**А.В. Тигушкин**



МИНОБРНАУКИ РОССИИ

федеральное государственное автономное  
образовательное учреждение высшего образования  
«Санкт-Петербургский национальный  
исследовательский университет  
информационных технологий,  
механики и оптики» (Университет ИТМО)

Кронверкский проспект, д. 49, г. Санкт-Петербург,  
Российская Федерация, 197101  
тел.: (812) 232-97-04 | факс: (812) 232-23-07  
od@mail.ifmo.ru | www.ifmo.ru

08.09.2015 № 4-25/1429

УТВЕРЖДАЮ

Проректор по ИР  
Университета ИТМО  
д.т.н., профессор  
В.С. Никифоров



### АКТ О ВНЕДРЕНИИ

результатов диссертационной работы  
на соискание ученой степени  
кандидата технических наук  
**Тесли Николая Николаевича**

Комиссия в составе: председателя — д-р техн. наук, проф. Парфенова В.Г.,  
заведующего кафедрой информационных систем; и членов комиссии:

- канд. физ.-мат. наук, Зубка Д.А., доцента кафедры информационных систем,
- канд. пед. наук, доц. Маятина А.В., доцента кафедры информационных систем,

Составила настоящий акт, подтверждающий, что результаты диссертационной работы Тесли Н.Н. «Разработка методов и моделей построения сервис-ориентированной системы обеспечения инфомобильности» используются в учебном процессе кафедры информационных систем федерального государственного автономного образовательного учреждения высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики».

Основные понятия, определения и результаты диссертационного исследования используются в дисциплине «Интеллектуальные системы и технологии» специальности 23040002 – «Информационные системы и технологии».

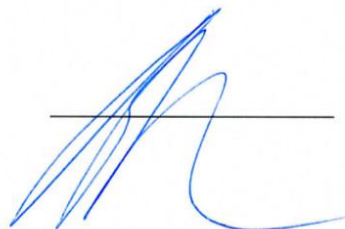
В данной дисциплине используются следующие результаты диссертационной работы:



1. Сервис-ориентированный подход к построению системы обеспечения инфомобильности, использующий технологию интеллектуальных пространств для обеспечения совместного доступа к информации несколькими сервисами.
2. Концептуальная, онтологическая и сценарная модели сервис-ориентированной системы обеспечения инфомобильности.
3. Модификация метода автоматического сопоставления онтологий, позволяющая использовать технологию краудсорсинга при сопоставлении онтологий.

Использование указанных результатов в учебном процессе позволило предоставить студентам актуальные знания о способах построения сервис-ориентированных систем, их возможных архитектурах на примере системы обеспечения инфомобильности. В материалах курса демонстрируются способы организации взаимодействия сервисов, применения онтологии для структурирования информации и обмена информацией между разнородными сервисами. Подробно описывается метод сопоставления онтологий, дается его анализ и возможное расширение с использованием технологии краудсорсинга.

Председатель  
Заведующий кафедрой ИС  
д-р техн. наук, проф. Парфенов В.Г.



Члены комиссии:

канд. физ.-мат. наук, Зубок Д.А.



канд. пед. наук, доц. Маятин А.В.





Министерство образования и науки Российской Федерации  
 Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)» (СПбГЭТУ)

УТВЕРЖДАЮ

Проректор по учебной работе,  
 канд. техн. наук, доц.

В. Н. Павлов

20 07 2015г.

### А К Т

О внедрении результатов диссертационной работы  
 на соискание ученой степени  
 кандидата технических наук  
**Тесли Николая Николаевича**

Комиссия в составе: председателя, и.о. заведующего кафедрой автоматизированных систем обработки информации и управления канд. техн. наук, проф. Цехановского В.В., и членов комиссии: канд. техн. наук, проф. Мустафина Н.А и д-р. техн. наук, проф. Водяхо А.И. составили настоящий акт о том, что результаты диссертационной работы «**Разработка методов и моделей построения сервис-ориентированной системы обеспечения инфомобильности**» использованы при проведении курса «Сервисно-ориентированные системы» магистерской программы 230400.68 «Информационные системы и технологии».

Использованы следующие результаты работы:

1. Подход к построению сервис-ориентированной системы обеспечения инфомобильности.
2. Концептуальная модель организации работы сервисов системы обеспечения инфомобильности
3. Онтологическая модель представления контекста пользователя.
4. Модификация метода сопоставления онтологий, использующая привлечение экспертов при невозможности автоматического сопоставления онтологий.

Использование указанных результатов в учебном процессе позволяет предоставить студентам актуальные знания о способах построения сервис-ориентированных систем, их возможных архитектурах. Демонстрируются способы организации взаимодействия сервисов с помощью онтологии. Подробно описывается метод сопоставления онтологий, дается его анализ и возможное расширение с использованием краудсорсинга.

Председатель комиссии

\_\_\_\_\_ канд. техн. наук, проф., Цехановский В.В.

Члены комиссии

\_\_\_\_\_ д-р. техн. наук, проф. Водяхо А.И.

\_\_\_\_\_ канд. техн. наук, проф. Мустафин Н.А.