

Метод онтолого-ориентированной разработки программных агентов с косвенным взаимодействием в интеллектуальных пространствах

Ломов Александр Андреевич

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

Научный руководитель:
к.ф.-м.н., доцент, Д. Ж. Корзун

Петрозаводский государственный университет
кафедра информатики и математического обеспечения

Интеллектуальные пространства

ИП обеспечивает множество участников контекстно–зависимой информацией, сервисами, персонифицированными рекомендациям, где и когда они требуются, по возможности в упреждающей манере.

Участники ИП — **агенты** (персональные, оборудования, сервисные и др.).

Взаимодействие для совместной обработки информации: формирование и получение контекста, извлечение знаний из данных. Программируется в логике агента – программный коде, реализующий действия агента в системе.

Особенности исследуемых ИП:

- локализованные Интернет-ориентированные вычислительные среды;
- взаимодействие агентов через разделяемое информационное содержимое;
- онтолого-ориентированные технологии Семантического веб: RDF, OWL, SPARQL.

Свойство	Подход / Модель		
Организация информационного содержимого	Индивидуальные хранилища участников	Общее информационное хранилище	
Сетевое взаимодействие	Обмен сообщениями	Публикация/подписка	Классная доска
	Прямое		Косвенное
Вычислительная среда	Глобально распределенная	Локализованная	Локальная
	Виртуальная	Физическая	Социально-ориентированная

Актуальность темы

Бурное развитие технологий и аппаратного обеспечения приводит к:

- разнообразию цифровой аппаратуры для применения в различных областях деятельности человека;
- средам повсеместных вычислений с массовым взаимодействием устройств и переходу к Интернету физических устройств (IoT);
- глобализации и существенному ускорению процессов накопления и обработки информации.

Требуются прикладные системы таких классов как “умный дом”, “интеллектуальный зал” (системы Z-Wave, KNX, C-Bus, СКАТ).

Разработка таких систем сталкивается с технологическими проблемами:

- программирование разнообразных агентов и их взаимодействия: сенсоры, умные IoT-объекты, программные агенты;
- совместная обработка неоднородного информационного содержимого из множества источников и различных проблемных областей;
- онтологическая модель проблемной области — информационные объекты для взаимодействия агентов и совместной обработки.

Цель, объект и предмет исследования

Цель: повышение эффективности разработки агентов с косвенным взаимодействием в интеллектуальных пространствах за счет автоматизированной онтолого-ориентированной разработки программного кода логики агента на основе онтологий проблемной области.

Объект: программные агенты ИП, применяющие

- 1) модели “классная доска”, “публикация/подписка” для косвенного взаимодействия с другими агентами,
- 2) RDF-модель и OWL-онтологии для представления и совместной обработки информационного содержимого.

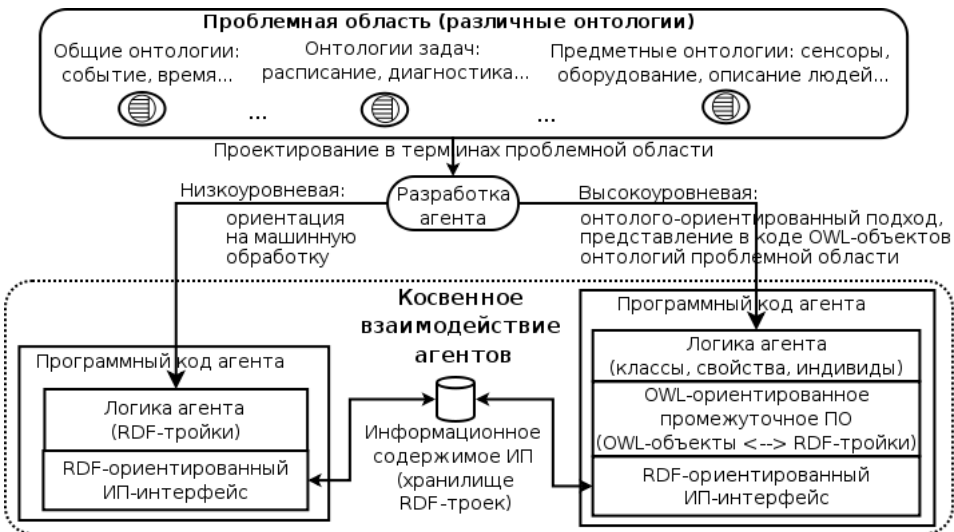
Предмет: модели косвенного взаимодействия агентов и механизмы программирования, обеспечивающие

- 1) автоматизированную онтолого-ориентированную разработку программного кода логики агента для разнообразных аппаратно-программных вычислительных платформ,
- 2) интеграцию, синхронизацию и обработку на стороне агента во время его выполнения информационного содержимого с использованием объектов онтологической модели проблемной области.

Положения выносимые на защиту (паспорт 05.13.11)

- 1** Метод автоматизированной онтолого-ориентированной разработки для программирования логики агента с использованием объектов онтологической модели проблемной области и моделей косвенного взаимодействия агентов (пп. 5, 9).
- 2** Модель взаимодействия агентов на основе сессии параллельных сеансов сетевого доступа для участия агента в нескольких ИП и автоматической интеграции на стороне агента объектов из этих ИП (п. 9).
- 3** Модель взаимодействия агентов на основе операции подписки для отслеживания агентом происходящих в ИП изменений и автоматической синхронизации на стороне агента измененных в ИП объектов (п. 9).
- 4** Модель взаимодействия агентов на основе обработки группы объектов на стороне агента с автоматическим формированием запроса на множественное изменение информационного содержимого ИП как транзакции (п. 9).
- 5** Механизмы обеспечения онтолого-ориентированной разработки программных агентов на основе предложенных моделей взаимодействия агентов и реализация этих механизмов в инструментарии SmartSlog платформы Smart-M3 (пп. 12, 16).

Реализация косвенного взаимодействия агентов в ИП

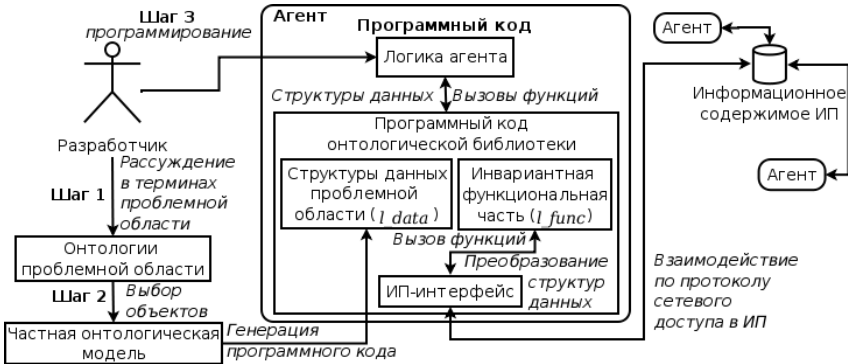


Задачи программирования логики агента

Задача	RDF-представление	OWL-представление	Автоматизация
Представление объектов проблемной области в программном коде.	Программный код оперирует наборами троек.	Структуры данных, приближенные к проблемной области.	Генерация промежуточного ПО для логики агента.
Сетевой доступ к нескольким ИП с интеграцией поступающих обновлений.	Сетевые сеансы доступа в параллельных вычислительных потоках с отдельной обработкой получаемых из ИП троек.	Сетевые сеансы объединены общим локальным хранилищем.	Интеграция поступающих обновлений из ИП.
Отслеживание изменений в ИП, вносимых другими агентами, и синхронизация изменений.	Подписка на тройки и программирование обработки получаемых наборов троек.	Подписка на онтологические объекты (классы, индивиды, свойства).	Синхронизация поступающих обновлений из ИП.
Локальная обработка с последующим внесением множественных изменений в ИП.	Обработка наборов троек и формирование множества запросов на внесение изменений.	Обработка в терминах проблемной области и формирование запросов на изменение индивидов и их свойств.	Формирование общего запроса на внесение изменений в ИП.

Промежуточное ПО — программная **онтологическая библиотека** функций, предоставляющая дополнительный уровень абстракции для работы с объектами онтологической модели на уровне логики агента.

Метод разработки для программирования логики агента



При разработке заданного агента выполняются следующие шаги.

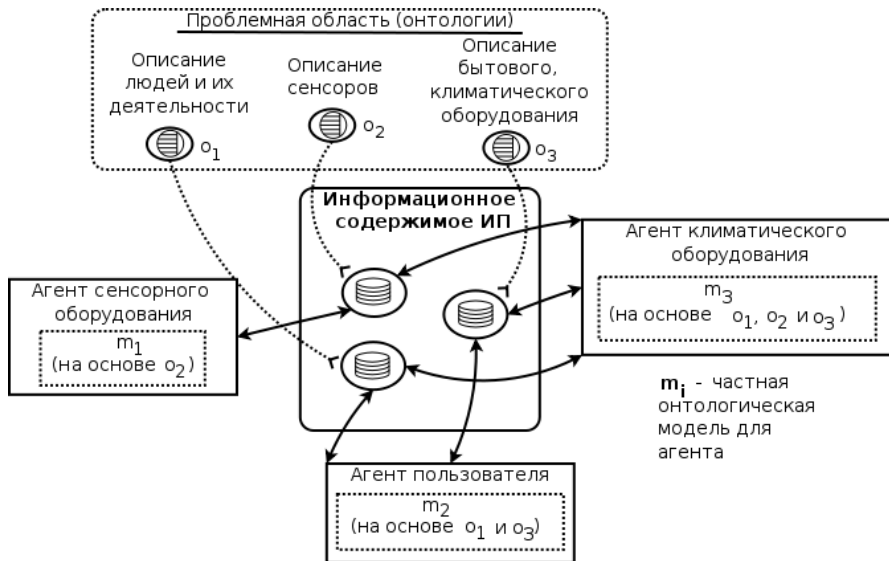
Шаг 1. Онтологическое моделирование проблемной области.

Шаг 2. Выбор объектов для взаимодействия в ИП и построение кода онтологической библиотеки:

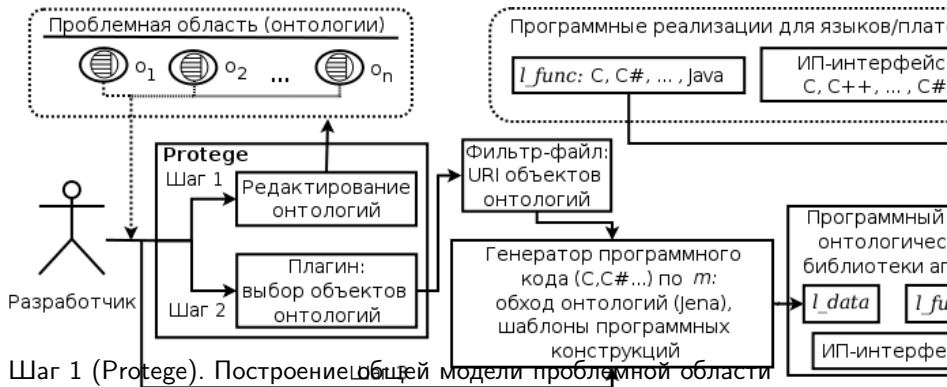
I_{data} — генерируется, I_{func} — определяется целевой платформой.

Шаг 3. Программирование логики агента с использованием функций и структур данных из онтологической библиотеки.

Пример “Умный дом”: онтологии проблемной области



Механизм генерации программного кода онтологической библиотеки



Шаг 1 (Protege). Построение общей модели проблемной области $O = o_1 + o_2 + \dots + o_n$.

Шаг 2 (разработан плагин для Protege). Формирование частной онтологической модели m : автоматизированный выбор объектов из O .

Шаг 3 (разработан генератор). Генерация программного кода (l_{data}): анализ OWL-онтологий (Jena), фильтр-файл, выбор языка программирования.

Модель взаимодействия агентов на основе сессии

При программировании логики агента модель позволяет:

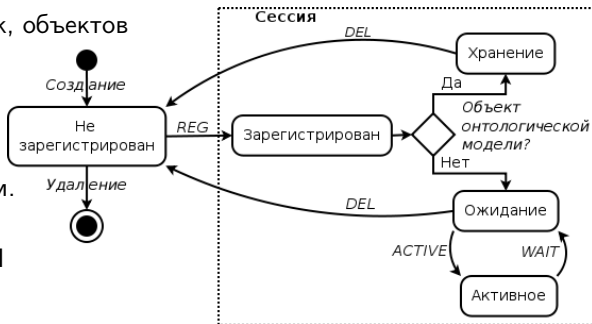
- объединять работу сеансов и подписок, участвуя в нескольких ИП;
- локально интегрировать объекты из нескольких ИП.

Агент использует модель как (P_s, Q_s, D_s) :

P_s и Q_s — множества сетевых сеансов и подписок на изменение объектов,
 D_s — множество объектов онтологической модели (локальное хранилище).

Состояния сеансов, подписок, объектов онтологической модели:

- **Зарегистрирован/не зарегистрирован** — привязка к сессии объекта, сеанса, подписки.
- **Активное/ожидание** — наличие соединения с ИП у сеанса, подписки.
- **Хранение** — объект в D_s .



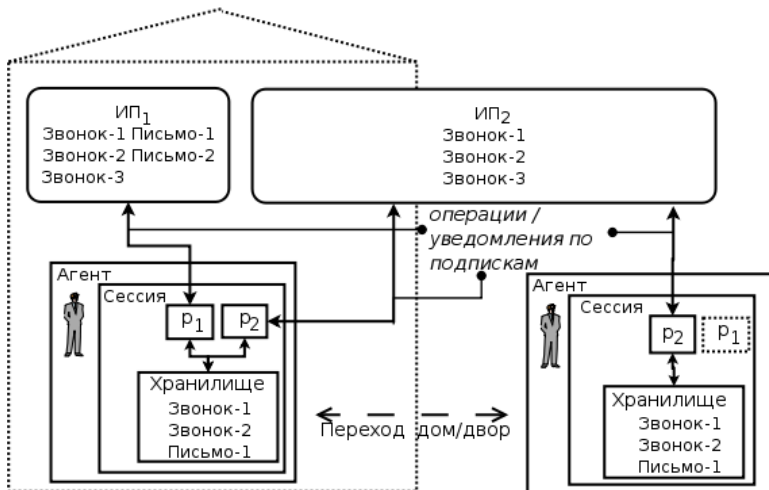
Переходы (операции управления): *REG*, *DEL*, *ACTIVE*, *WAIT*.

Пример “Умный дом”: работа с двумя ИП

Внутри дома — ИП₁, во дворе — ИП₂.

Агент использует сеансы p_1 и p_2 .

В доме активны и p_1 , и p_2 ; во дворе — только p_2 .



Механизм программирования логики агента на основе модели сессии

- Агент объединяет несколько сетевых сеансов в одну сессию.
- Запросы к ИП используют зарегистрированные в сессии объекты онтологической модели.
- Агент может создать несколько сессий.

p_1 – в доме.
 p_2 – в доме и во дворе.
 $REG(p_1)$, $REG(p_2)$,
 $REG(Call)$, $REG(Mail)$



Переход:

- во дворе
 $WAIT(p_1)$,
 $ACTIVE(p_2)$
- в доме
 $ACTIVE(p_1)$,
 $ACTIVE(p_1)$

$WAIT(p_1)$, $WAIT(p_2)$,
 $DEL(p_1)$, $DEL(p_2)$

Модель взаимодействия агентов на основе подписки

При программировании логики агента модель позволяет:

- отслеживать изменения объектов ИП;
- синхронизировать локальные объекты с изменениями в ИП.

Агент использует модель как $(O_{src}, D_{sub}, D_{ind})$:

O_{src} – множество объектов для отслеживания изменений,

D_{sub} – множество RDF-представлений отслеживаемых объектов,

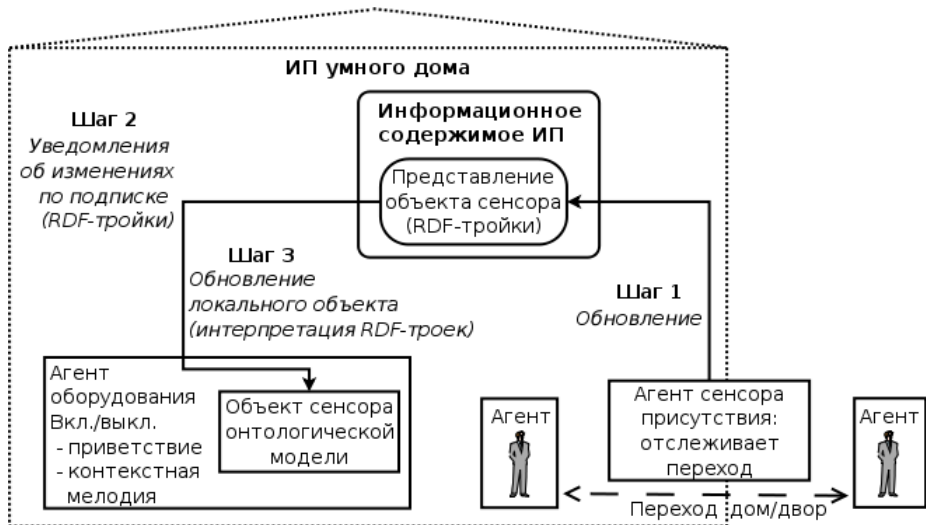
D_{ind} – множество RDF-представлений поступающих изменений.

Трансляция отслеживаемых объектов $SUBDATA : O_{src} \rightarrow D_{sub}$.

Обновление $UPDATE : D_{ind} \rightarrow O_{src}$, где $|D_{ind}| \gg |O_{src}|$.



Пример “Умный дом”: отслеживание присутствия людей



Механизм программирования логики агента на основе модели онтолого-ориентированной операции подписки

O_{src} — множество объектов онтологической модели:
классы и пары “Индивид – список свойств”.

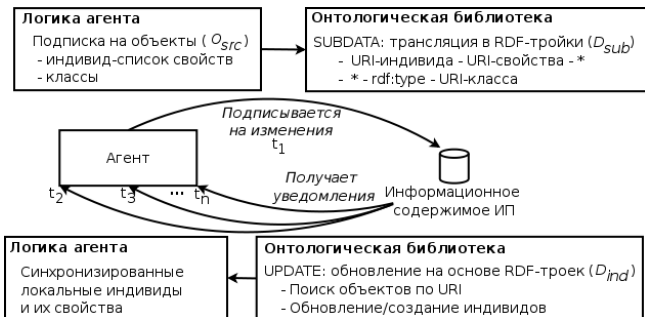
D_{sub} — множество шаблонов RDF-троек (с маской “*”).

D_{ind} — множество RDF-троек.

Дискретные моменты времени:

t_1 — подписывание на изменения (разовая операция);

$t_2, t_3 \dots t_n$ — получение уведомлений об изменениях (агент ожидает).



Модель взаимодействия агентов на основе обработки группы объектов

При программировании логики агента модель позволяет:

- определять правило обработки для группы объектов;
- объединять операции к ИП.

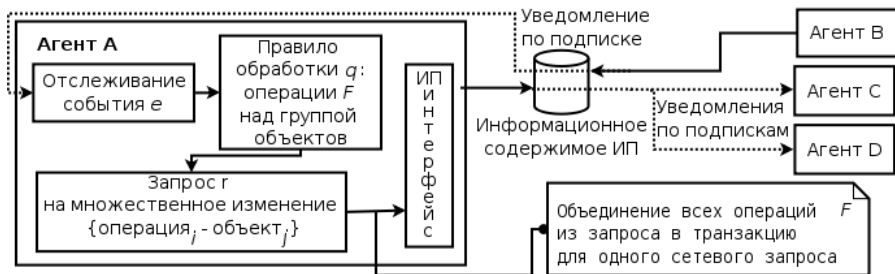
Агент использует модель как (e, F, q, r) :

e — отслеживаемое событие (вручную, по подписке),

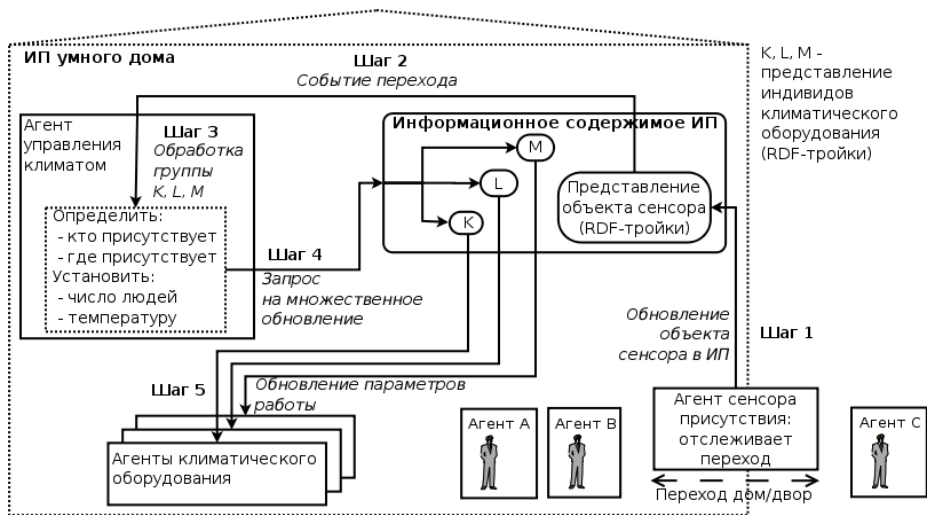
F — множество операций над объектами и их значениями,

q — правило обработки, привязывающее операции к объектам,

r — набор {операция–объект} (формируемый запрос на обработку).



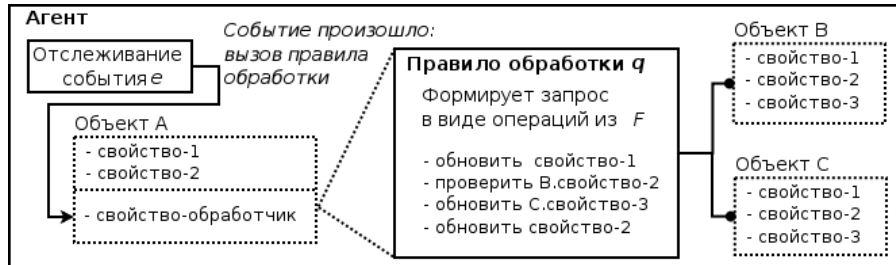
Пример “Умный дом”: управление климатом в зависимости от присутствия людей



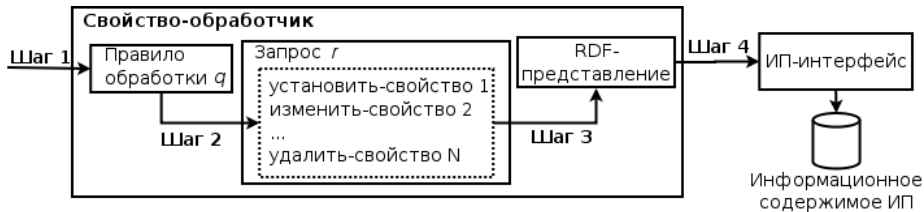
Механизм программирования логики агента на основе модели для обработки группы объектов

Свойство-обработчик: программная конструкция для агента, вводящая дополнительное свойство индивида онтологической модели.

- Работа как со свойством индивида онтологической модели.
- Определение правила обработки q для группы объектов.
- Значение свойства-обработчика — событие e .
- Наступление события — обновление свойства-обработчика.



Обработка группы объектов во время выполнения программы агента



Шаг 1. Вызов правила обработки q при помощи обновления свойства-обработчика.

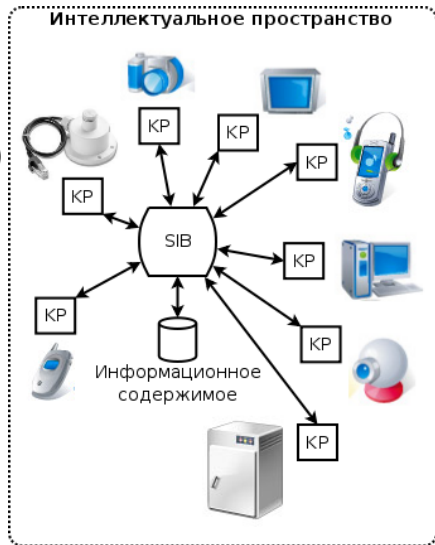
Шаг 2. Формирование правилом q запроса r на основе операций $F = \{INS, DEL, UPD, QRY\}$ (установить, удалить, обновить, получить).

Шаг 3. Запрос r преобразуется в RDF-формат ИП-интерфейса.

Шаг 4. Выполнение запроса в виде одной сетевой транзакции к ИП, объединяющей все операции из r .

Платформа Smart-M3

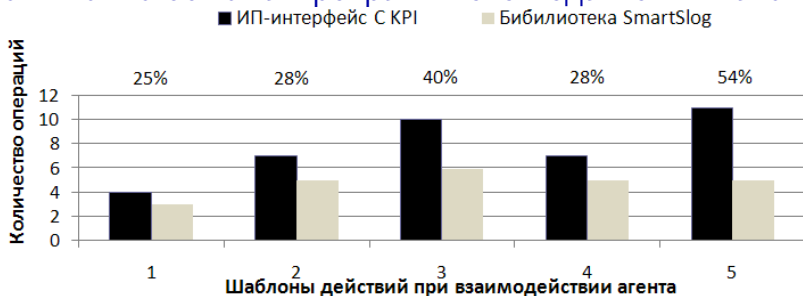
- Развертывание ИП в заданной IoT-среде.
- Агенты – процессоры знаний (KP) на различных устройствах.
- Семантический информационный брокер (SIB) реализует доступ к общему информационному содержимому — хранилищу RDF-троек.
- Протокол доступа к ИП:
 - сетевые сеансы между KP и SIB;
 - RDF-тройки — параметры операций доступа (запросов).
 - Операции доступа: вход/выход в/из ИП, поместить/удалить/обновить RDF-тройки и др.



Инструментарий SmartSlog

- Разработка агентов для ИП на платформе Smart-M3:
 - ▶ генерация онтологических библиотек (библиотек SmartSlog) на основе OWL-онтологий проблемной области и выбора объектов;
 - ▶ программирование логики агента в OWL-подобных терминах;
 - ▶ поддержка различных ИП-интерфейсов и языков программирования.
- Взаимодействие агентов на основе сессии:
 - ▶ управление агентом состоянием сессии, ее сеансов и подписок;
 - ▶ параллельный доступ агента к нескольким ИП;
 - ▶ интеграция на стороне агента объектов из нескольких ИП.
- Взаимодействие агентов на основе онтолого-ориентированной операции подписки:
 - ▶ синхронный и асинхронный способы обработки уведомлений;
 - ▶ подписка на свойства и классы индивидов.
- Взаимодействие агентов на основе обработки группы объектов:
 - ▶ обработка агентом группы объектов локально;
 - ▶ запрос на множественное изменение объектов в ИП — транзакция.

Уменьшение объема программного кода логики агента



1. Получение свойств индивида из одного ИП.
2. Получение свойств индивида из нескольких ИП.
3. Синхронная подписка на изменение свойств индивида в ИП.
4. Асинхронная подписка в параллельном потоке на свойства индивида в ИП (вариант с поддержкой в ИП-интерфейсе).
5. Асинхронная подписка в параллельном потоке на свойства индивида в ИП (вариант без поддержки в ИП-интерфейсе).

Уменьшение числа программируемых разработчиком операций:
в среднем на 35%.

Упрощение программного кода логики агента

Пример Hello World: отслеживание “приветствия” (свойство индивида):

- без подписки — циклический запрос на получение свойства из ИП;
- синхронная подписка — ожидание уведомлений об изменениях в ИП.

ИП-интерфейс С КРІ — разработчик программирует обработку уведомлений (проверка статуса операции, обработка ошибок) и завершение подписки.

Пример GPS: отслеживание появления индивидов и их координат.

Оцениваются возможности библиотеки SmartSlog:

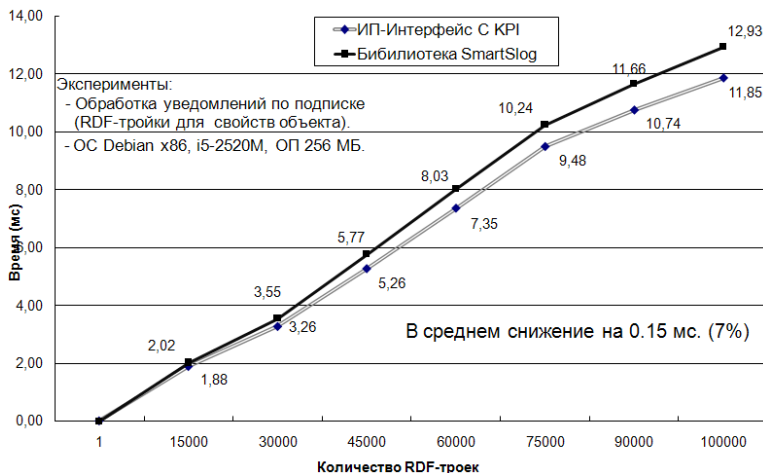
- асинхронная подписка на класс индивидов;
- восстановление: уведомление о сбое сетевого соединения с ИП.

Вариант реализации примера	С КРІ		SmartSlog		
	Hello World		Hello World	GPS	
	Без подписки	Синхронная подписка	Синхронная подписка	Асинхронная подписка	Восстановление соединения
Строк кода	49	144	68	173	194
Цикломатическая сложность (С)	13	37	19	38	40

Снижение С: упрощение тестирования (меньше тестов для покрытия кода).

Анализ снижения производительности

- ИП-интерфейс С КРІ (сам разработчик):
хранение списка RDF-троек, проверка дублирования
- SmartSlog (автоматически):
создание, обновление, проверка дублирования объектов



Анализ увеличения потребления памяти

На стороне агента основное потребление:

- хранение структур данных для индивидов и значений свойств;
- Создание и хранение индивида: 48 байт + данные.
- Создание и хранение значения свойства индивида: 20 байт + данные.
- Данные из структур RDF-троек, структуры удаляются после обработки.

Класс агентов	Пример устройства	Кол-во объектов	Макс. доп. объем памяти (Кб)	
			Индивиды	Свойства
Агенты сенсоров	Одноплатные компьютеры с сенсорами.	до 50	2,344	0,977
Агенты оборудования (встроенными ЭВМ)	Холодильники, телевизоры, медиа-доски и т.д.	до 100	4,688	1,953
Персональные мобильные устройства	Телефон, нетбук, планшет.	до 1000	46,875	19,531
Сервисы, сервис-посредники для организации работы агентов	Настольные, серверные ЭВМ.	от 10000	от 468,75	от 195,312

Стоимость памяти снижается даже для малопроизводительных устройств.

Агент локально работает с частью объектов: динамическая загрузка из ИП.

Обработка информации от маломощных сенсоров через посредника (одноплатный компьютер).

Внедрение и эффективность использования

Применение инструментария SmartSlog при разработке системы интеллектуального зала SmartRoom (проведение конференций, совещаний).

Возможности	Результат применения
Поддержка различных вычислительных устройств.	Агенты работают на телефонах, планшетах, одноплатных компьютерах с сенсорами (Raspberry Pi), мультимедийных досках.
Частные онтологические модели, плагин Protege для выбора объектов.	Генерация библиотек SmartSlog из набора OWL-онтологий для описания сервисов, людей-участников, IoT-объектов.
Использование сессии для параллельных сеансов и подписок.	Переключение состояния сеансов/подписок при вкл/выкл устройства (напр., мобильного телефона), отслеживание ошибок соединения сеанса, подписки с ИП.
Обработка группы объектов.	Объединение запросов к ИП для уменьшения сетевой нагрузки (напр., пересчет времени выступления для участников).
Разработка в терминах предметной области.	Уменьшение кода в логике агента для взаимодействия в ИП (напр., подписка для отслеживания слайда или выступающего).

Возможности разработки для различных аппаратно-программных вычислительных платформ

- Поддержка библиотек SmartSlog для языков ANSI C и C#.
- Стандарт ANSI поддерживается многими платформами
- Малое число зависимостей от сторонних компонент.
- Подключение ИП-интерфейсов через программный адаптер.

Экспериментальная апробация для платформ:

Платформа / Язык	Windows	Windows Phone	Linux семейство	Mac OS	Android
ANSI C (C KPI)	+	-	+	+	+ (взаимодействие ANSI C и Java кода)
C# (C KPI)	+ (C KPI адаптер)	+ (C KPI адаптер)	На базе Mono		-
C# (C# KPI)	+	+			-

Метрики кода

Инвариантная функциональная часть I_{func} библиотеки SmartSlog

ANSI C	C#
Строк кода: 6073	Строк кода: 2247
Вклад автора: 50%	Вклад автора: 90%

Поддержка ИП-интерфейсов

Базовый C# KPI адаптер	C KPI адаптер (Windows)	C KPI адаптер (Windows Phone)
Строк кода: 134	Строк кода: 607	Строк кода: 257
Вклад автора: 100%	Вклад автора: 100%	Вклад автора: 100%

Плагин Protege для отбора объектов онтологии (Java)

- Строк кода: 484
- Вклад: 100%

Открытый код (GPL v2): <http://sourceforge.net/projects/smartslog/>

Результаты диссертационной работы

- 1** Разработан метод автоматизированной разработки агентов, отличающийся выбором объектов онтологической модели для косвенного взаимодействия в ИП и позволяющий генерировать код онтологической библиотеки на целевом языке программирования
- 2** Разработана модель взаимодействия агентов на основе сессии, отличающаяся поддержкой параллельных сеансов сетевого доступа и позволяющая агенту интегрировать объекты из нескольких ИП.
- 3** Разработана модель взаимодействия агентов на основе операции подписки, отличающаяся возможностью отслеживания изменений объектов в ИП и позволяющая агенту их синхронизировать.
- 4** Разработана модель взаимодействия агентов на основе обработки группы объектов, отличающаяся возможностью определять правило обработки объектов как реакцию на событие и позволяющая агенту вносить множественные изменения в ИП как транзакцию.
- 5** Реализованы механизмы онтолого-ориентированной разработки агентов в виде инструментария SmartSlog, отличающиеся поддержкой генерации программного кода онтологических библиотек и обеспечивающие разработку программных агентов на платформе Smart-M3.

Публикации и апробация результатов

10 публикаций (из них 3 ВАК, 1 Scopus)

- Корзун Д.Ж. Автоматизированная модельно-ориентированная разработка программных агентов для интеллектуальных пространств на платформе Smart-M3 / Д. Ж. Корзун, **А. А Ломов**, П. И. Ваняг // Теоретический и прикладной научно-технический журнал Программная инженерия. №5. 2012. С. 6-14.
- **Ломов А. А.** Операция подписки для приложений в интеллектуальных пространствах платформы Smart-M3 / А. А. Ломов, Д. Ж. Корзун // Труды СПИИРАН. Вып. 4(23). 2012. С.439-458.
- **Ломов А. А.** Взаимодействие программного агента на уровне сессии с интеллектуальным пространством // Ученые записки ПетрГУ. Вып. 10 (139). 2013.

5 международных конференций.

Внедрение

ПетрГУ

Система интеллектуального зала SmartRoom реализуется в ИТ-Парке ПетрГУ в аудиториях 104 и 403.

Курс лекций "Интеллектуальные сетевые пространства".

Практические занятия с использованием платформы Smart-M3 и инструментария SmartSlog.

Конференции Ассоциации Открытых Инноваций FRUCT

Система интеллектуального зала SmartRoom использовалась для проведения секций этих конференций:

- FRUCT 12 (5-9 ноября 2012, г. Оулу)
секции "Internet of Things and Smart Spaces" и "Smart Space Technologies".
- FRUCT 13 (22-26 апреля 2013, г. Петрозаводск)
секция "Internet of Things and Smart Spaces I".
- FRUCT 14 (11-15 ноября 2013, г. Хельсинки)
Демо-секция.