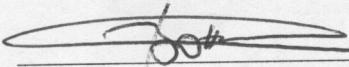


Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

УДК 004.04, 004.82, 004.7  
№ госрегистрации 114121750065  
Инв. №

УТВЕРЖДАЮ  
Проректор по научной работе  
д.т.н., профессор

 В.О. Никифоров

«30» декабря 2014 г.

ОТЧЕТ  
О ПРИКЛАДНЫХ НАУЧНЫХ ИССЛЕДОВАНИЯХ

по теме:

Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)

Этап 1 «Выбор направления исследований»  
(промежуточный)

Шифр № 340745

Руководитель ПНИ  
к.т.н, доцент

 Д.И. Муромцев  
(подпись, дата) 30.12.14

Санкт-Петербург 2014

# СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель темы:

к.т.н., доцент

Д.И. Муромцев

(подпись, дата) 30.12.14 (введение, заключение)

Исполнители темы:

аспирант

М.А. Колчин

(раздел 1, раздел 2)

аспирант

О.В. Пархимович

(раздел 1, раздел 4)

аспирант

Р.Я. Лабковская

(раздел 2)

д.т.н., профессор

С.А. Арутамов

(раздел 1, раздел 2)

инженер

С.О. Попов

(раздел 3)

к.т.н., доцент

И.А. Радченко

(раздел 4)

лаборант-исследователь

27.12.14

(подпись, дата)

Д.С. Павлов

(раздел 2, раздел 5)

лаборант-исследователь

27.12.2014

(подпись, дата)

Ю.В. Емельянов

(раздел 2, раздел 5)

зав. кафедры

30.12.14

(подпись, дата)

О.А. Кураш

(раздел 1, раздел 3)

начальник отдела

30.12.14

(подпись, дата)

Г.Л. Маркина

(раздел 5)

к.т.н., старший научный  
сотрудник

30.12.14

(подпись, дата)

С.И. Баландин

(раздел 2, раздел 4)

к.т.н., доцент

30.12.14

(подпись, дата)

Д.А. Заколдаев

(раздел 2, приложение А)

ведущий инженер

30.12.14

(подпись, дата)

С.А. Середа

(раздел 3)

аналитик отдела  
разработки ПО

30.12.14

(подпись, дата)

И.А. Филькова

(раздел 3)

руководитель отдела  
разработки

30.12.14

(подпись, дата)

Д.О. Захаров

(раздел 3)

Нормоконтролер:

  
Подпись, дата)

В.В. Беззубик

## РЕФЕРАТ

Отчет 303 с., 39 рис., 21 табл., 7 прил., 1 ч., 170 источников.

СЕМАНТИЧЕСКИЙ ИНТЕРНЕТ ВЕЩЕЙ, СЕМАНТИЧЕСКИЕ ТЕХНОЛОГИИ, ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ, ОНТОЛОГИИ, ОБРАБОТКА СЛОЖНЫХ СОБЫТИЙ, СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА, БЕСПРОВОДНЫЕ СЕНСОРНЫЕ СЕТИ, ОБРАБОТКА ПОТОКОВЫХ RDF ДАННЫХ, ИНТЕРНЕТ ВЕЩЕЙ, ВИЗУАЛИЗАЦИЯ ГЕТЕРОГЕННЫХ ДАННЫХ, ИНТЕГРАЦИЯ ГЕТЕРОГЕННЫХ ДАННЫХ, АНАЛИЗ ПОТОКОВЫХ ДАННЫХ, АНАЛИЗ ГЕТЕРОГЕННЫХ ДАННЫХ, ПУБЛИКАЦИЯ ПОТОКОВЫХ ДАННЫХ.

В отчете представлены результаты исследований, выполненных по этапу 1 ПНИ «Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)».

Целью работы по данному этапу прикладного научного исследования является выбор направления исследований.

Общей целью выполнения ПНИ является:

- а) создание комплекса научных/научно-технических решений в области разработки методов и алгоритмов, обеспечивающих повышение эффективности научных исследований посредством агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things).

б) получение значимых научных результатов в области разработки масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things).

Результаты, полученные в рамках первого этапа, включают в себя:

- а) аналитический обзор современной научно-технической, нормативной, методической литературы, затрагивающей научно-техническую проблему, исследуемую в рамках ПНИ, в том числе обзор научных информационных источников: статьи в ведущих зарубежных и (или) российских научных журналах, монографии и (или) патенты) - не менее 15 научно-информационных источников за период 2009 - 2014 гг.
- б) патентные исследования в соответствии с ГОСТ Р 15.011-96 (см. Приложение А-Г).
- в) выбор направления исследований в области разработки масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things).
- г) разработка ПАС с подсистемой виртуальных серверов, подсистемой интеграции больших массивов гетерогенных данных и симуляцией распределенных вычислений для моделирования сети электронных потребительских устройств (Internet of Things).

- д) анализ существующих приемов онтологического инжиниринга и технологий семантического веба для агрегации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных.
- е) анализ применимости технологии связанных данных (Linked Data) для задачи интерактивной визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	22
1 Аналитический обзор современной научно-технической, нормативной, методической литературы, затрагивающей научно-техническую проблему, исследуемую в рамках ПНИ, в том числе обзор научных информационных источников .....	24
1.1 Предпосылки созданию распределенной сети электронных потребительских устройств (Internet of Things) .....	24
1.2 Определение распределенной сети электронных потребительских устройств (Internet of Things) .....	26
1.3 Элементы технологий Интернета вещей.....	29
1.3.1 Краткий обзор элементов технологий Интернета вещей.....	29
1.3.2 Технология радиочастотной идентификации (RFID).....	29
1.3.3 Технологии беспроводных сенсорных сетей (WSN).....	30
1.3.4 Методы адресации электронных потребительских устройств .....	32
1.3.5 Методы хранения и анализа данных .....	33
1.3.6 Методы и технологии визуализации данных .....	34
1.4 Обзор тенденций в развитии технологий Интернета вещей .....	35
1.5 Обзор существующих исследовательских проблем .....	39
1.5.1 Основные исследовательские проблемы .....	39
1.5.2 Архитектура Интернета вещей .....	40
1.5.3 Адресация и присваивание имен .....	41
1.5.4 Автоматическое конфигурирование электронных потребительских устройств .....	41
1.5.5 Стандартизация .....	42

1.5.6	Контекстно-зависимые вычисления .....	42
1.5.7	Методы поиска и отбора сенсоров .....	43
1.5.8	Безопасность, конфиденциальность и доверие .....	43
1.6	Обзор областей применения технологий Интернета вещей .....	44
1.6.1	Классификация уровней областей применения .....	44
1.6.2	Области персонального и домашнего применения .....	45
1.6.3	Области применения на уровне предприятия .....	46
1.6.4	Области применения на уровне района или города .....	48
1.6.5	Области широкого применения .....	49
2	Выбор направления исследований в области разработки масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений.....	51
2.1	Основные направления исследований.....	51
2.2	Методы и алгоритмы агрегации больших массивов гетерогенных данных с применением приемов онтологического инжиниринга и технологий семантического веба .....	52
2.2.1	Основные протоколы агрегации данных Интернета вещей .....	52
2.2.2	Протокол передачи данных прикладного уровня CoAP (Constrained Application Protocol) .....	54
2.2.3	Обзор реализаций протокола CoAP .....	56
2.3	Методы и алгоритмы нормализации и анализа больших массивов гетерогенных данных генерируемых сетью электронных потребительских устройств.....	59
2.3.1	Методы нормализации и анализа сенсорных данных .....	59
2.3.2	Моделирование сенсорных данных с помощью онтологий .....	60

2.4 Методы и алгоритмы интерактивной визуализации больших массивов гетерогенных данных .....	66
2.5 Методы передачи данных от потребительских устройств к центрам обработки данных.....	77
2.5.1    Обзор протоколов передачи данных .....	77
2.5.2    Поддерживаемые возможности протоколов передачи данных.....	78
2.6 Технические принципы и подходы к организации защиты информации передаваемой в распределенной сети электронных потребительских устройств (Internet of Things) .....	81
2.6.1    Обзор методов обеспечения безопасности в распределенной сети электронных потребительских устройств.....	81
2.6.2    Анализ моделей угроз и атак .....	84
2.6.3    Конкретные проблемы и потенциальные решения .....	85
3 Разработка программно-аппаратного стенда с подсистемой виртуальных серверов, подсистемой интеграции больших массивов гетерогенных данных и симуляцией распределенных электронных потребительских устройств (Internet of Things).....	90
3.1 Назначение программно-аппаратного стенда.....	90
3.2 Состав программно-аппаратного стенда.....	90
3.3 Аппаратное обеспечение.....	91
3.3.1    Процессор.....	91
3.3.2    Материнская плата .....	93
3.3.3    Оперативная память .....	94
3.3.4    Жесткий диск .....	95
3.3.5    Соответствие аппаратной части требованиям технического задания ...	96
3.4 Программное обеспечение.....	97

3.4.1	Архитектура программного обеспечения ПАС .....	97
3.4.2	Подсистема виртуальных серверов .....	98
3.4.3	Подсистема симуляции распределенных электронных потребительских устройств (Internet of Things) .....	124
3.4.4	Подсистема хранения текущих показаний моделируемых электронных потребительских устройств.....	132
3.4.5	Подсистема интеграции больших массивов гетерогенных данных ....	135
3.5	Результаты .....	142
4	Анализ существующих приемов онтологического инжиниринга и технологий семантического веба для агрегации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных.	144
4.1	Обзор существующих трендов .....	144
4.2	Типовая архитектура программного обеспечения на основе данных электронных потребительских устройств.....	145
4.3	Существующие исследовательские проблемы .....	146
4.4	Применение методов онтологического инжиниринга и технологий семантического веба .....	147
5	Анализ применимости технологии связанных данных (Linked Data) для задачи интерактивной визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных .....	151
5.1	Вызовы современности, связанные с возрастающим объемом данных .....	151
5.1.1	Источники возрастающих объемов данных .....	151
5.1.2	Данные, порождаемые человеком .....	154
5.1.3	Данные, порождаемые машинами .....	155
5.1.4	Обобщение проблематики.....	158
5.2	Способы решения проблемы увеличивающихся объемов данных .....	158

5.2.1	Конвенциональные СУБД .....	158
5.2.2	Специализированные СУБД .....	160
5.2.3	Big Data.....	162
5.2.4	Linked Data .....	163
5.3	Проблема разнородности данных, поступающих от электронных потребительских устройств.....	167
5.3.1	Синхронность и асинхронность.....	167
5.3.2	Различные форматы .....	168
5.3.3	Различная дискретность и задержки .....	168
5.3.4	Применимость технологий связанных данных (Linked Data) для решения проблемы .....	169
5.4	Проблема постоянно меняющейся среды .....	169
5.4.1	Нормативно-правовое поле .....	169
5.4.2	Смена технологий, форматов, аппаратного обеспечения .....	170
5.4.3	Запросы пользователей и новый функционал.....	170
5.5	Решение проблемы обработки данных в реальном времени .....	171
5.6	Решение задачи визуализации больших массивов данных в современных платформах.....	172
	<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>175</b>
	<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>	<b>181</b>
	Приложение А Отчет о патентных исследованиях.....	202
	Приложение Б Задание №1-IOT/14 на проведение патентных исследований.....	221
	Приложение В Регламент поиска № PAT-1-IOT/14 .....	223
	Приложение Г Отчет о поиске .....	226
	Приложение Д Руководство по эксплуатации ПАС .....	242

Приложение Е Структурная схема ПАС.....	251
Приложение Ж Формуляр .....	252
Приложение И Описание программного обеспечения ПАС .....	275
Приложение К Текст программного обеспечения ПАС .....	288

## ОПРЕДЕЛЕНИЯ

В настоящем отчете о ПНИ применяют следующие термины с соответствующими определениями:

Интернет вещей (Internet of Things) – концепция сети однозначно идентифицируемых встраиваемых вычислительных устройств в рамках существующей инфраструктуры Интернета.

Технологии семантического веба (Semantic Web technologies) – технологии, которые предоставляют общий инструмент, позволяющий осуществлять обмен данными и их многократное использование между и за пределами программ, предприятий и сообществ.

Связанные данные (Linked Data) – методы публикации взаимосвязанных между собой наборов структурированных данных в рамках существующей инфраструктуры Интернета.

Межмашинное взаимодействие (Machine-to-Machine, M2M) – общее название технологий, которые позволяют машинам по беспроводным или проводным каналам связи обмениваться информацией друг с другом, или же передавать её в одностороннем порядке.

Беспроводная сенсорная сеть (Wireless Sensor Network) – распределенная, самоорганизующаяся сеть множества датчиков (сенсоров) и исполнительных устройств, объединенных между собой посредством радиоканала.

Сервис-ориентированная архитектура (Service Oriented Architecture) – модульный подход к разработке программного обеспечения, основанный на использовании распределенных, слабо связанных заменяемых компонентов, оснащенных стандартизованными интерфейсами для взаимодействия по стандартизованным протоколам.

Atlas — компьютер, созданный в Великобритании совместно Манчестерским университетом Виктории и компаниями Ferranti и Plessey. Atlas являлся одним из первых суперкомпьютеров, и на момент создания был одним из самых производительных компьютеров в мире. Первый экземпляр Atlas был собран и официально введен в эксплуатацию в Манчестерском университете в 1962 году.

OpenVMS (англ. Open Virtual Memory System, или просто VMS) — проприетарная серверная операционная система, разработанная во второй половине 1970-х годов компанией Digital Equipment Corporation для серии компьютеров VAX. Позже портирована на платформы DEC Alpha и Intel Itanium. В августе 2014 года выкуплена у компании Hewlett-Packard и полностью принадлежит компании VMS Software Inc. Применяется для построения отказоустойчивых систем высокой готовности и mission-critical применений. Среди российских заказчиков OpenVMS преобладают оборонные структуры и банки, телекоммуникационные компании, предприятия непрерывного цикла (АЭС).

VM (англ. Virtual Machine) – операционная система для мейнфреймов фирмы IBM. Известна в русскоязычной литературе по названию её клона времен СССР – CBM (Система виртуальных машин). VM была разработана на основе более ранней ОС IBM CP/CMS, имевшей статус исследовательского проекта и разрабатывавшейся с 1967 года на базе модифицированной IBM System/360 Model 40, затем IBM System/360 Model 67 и IBM System/370, то есть систем, в которых впервые в семействе была реализована виртуальная память. Первая версия собственно VM, VM/370, была выпущена в 1972 году для семейства компьютеров IBM System/370. После этого был выпущен ряд версий VM для последующих семейств майнфреймов IBM: VM/ESA, VM/SEPP, VM/BSEPP, VM/SP, VM/HPO, VMXA/SF, VMXA/SP и, наконец, выпускающаяся с 2000-х годов – z/VM для 64-разрядного семейства IBM System z.

Веб-сервер — сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными. Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает. Клиент, которым обычно является веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы — это HTML-страницы, изображения, файлы, медиа-потоки или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

Виртуальная память (англ. *virtual memory*) — технология управления памятью ЭВМ, разработанная для многозадачных операционных систем. При использовании данной технологии для каждой программы используются независимые схемы адресации памяти, отображающиеся тем или иным способом на физические адреса в памяти ЭВМ. Позволяет увеличить эффективность использования памяти несколькими одновременно работающими программами, организовав множество независимых адресных пространств (англ.), и обеспечить защиту памяти между различными приложениями. Также позволяет программисту использовать больше памяти, чем установлено в компьютере, за счет откачки неиспользуемых страниц на вторичное хранилище (см. Подкачка страниц). При использовании виртуальной памяти упрощается программирование, так как программисту больше не нужно учитывать ограниченность памяти, или согласовывать использование памяти с другими приложениями. Для программы выглядит доступным и непрерывным все допустимое адресное пространство, вне зависимости от наличия в ЭВМ соответствующего объема ОЗУ.

Гетерогенность — разнородность, наличие неодинаковых частей в структуре, в составе чего-либо.

Майнфрейм (также майнфрейм, от англ. mainframe) — большой универсальный высокопроизводительный отказоустойчивый сервер со значительными ресурсами ввода-вывода, большим объёмом оперативной и внешней памяти, предназначенный для использования в критически важных системах (англ. mission-critical) с интенсивной пакетной и оперативной транзакционной обработкой. Основной разработчик майнфреймов — корпорация IBM, самые известные майнфреймы ей выпущены в рамках продуктовых линеек System/360, 370, 390, zSeries. В разное время майнфреймы производили Hitachi, Bull, Unisys, DEC, Honeywell, Burroughs, Siemens, Amdahl, Fujitsu, в странах СЭВ выпускались майнфреймы ЕС ЭВМ. Современные майнфреймы перестали быть закрытой платформой: они способны поддерживать на одной машине сотни серверов с различными операционными системами.

OLAP (англ. online analytical processing, аналитическая обработка в реальном времени) – технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу. Реализации технологии OLAP являются компонентами программных решений класса Business Intelligence.

OLAP-куб – OLAP-структура, созданная из рабочих данных. Куб создаётся из соединения таблиц с применением схемы звезды или схемы снежинки. В центре схемы звезды находится таблица фактов, которая содержит ключевые факты, по которым делаются запросы. Множественные таблицы с измерениями присоединены к таблице фактов. Эти таблицы показывают, как могут анализироваться агрегированные реляционные данные. Количество возможных агрегирований определяется количеством способов, которыми первоначальные данные могут быть иерархически отображены.

Песочница - жёстко контролируемый набор ресурсов для исполнения гостевой программы — например, место на диске или в памяти. Доступ к сети, возможность сообщаться с главной операционной системой или считывать информацию с

устройств ввода обычно либо частично эмулируют, либо сильно ограничивают. Песочницы представляют собой пример виртуализации. Повышенная безопасность исполнения кода в песочнице зачастую связана с большой нагрузкой на систему — именно поэтому некоторые виды песочниц используют только для неотлаженного или подозрительного кода. Как правило, песочницы используют для запуска непроверенного кода из неизвестных источников, как средство проактивной защиты от вредоносного кода, а также для обнаружения и анализа вредоносных программ. Также зачастую песочницы используются в процессе разработки программного обеспечения для запуска «сырого» кода, который может случайно повредить систему или испортить сложную конфигурацию. Такие «тестировочные» песочницы копируют основные элементы среды, для которой пишется код, и позволяют разработчикам быстро и безболезненно экспериментировать с неотлаженным кодом.

Проприетарное программное обеспечение (англ. proprietary software; от *proprietary* — частное, патентованное, в составе собственности и *software* — программное обеспечение) — программное обеспечение, являющееся частной собственностью авторов или правообладателей и не удовлетворяющее критериям свободного ПО (наличия открытого программного кода недостаточно). Правообладатель проприетарного ПО сохраняет за собой монополию на его использование, копирование и модификацию, полностью или в существенных моментах. Обычно проприетарным называют любое несвободное ПО, включая полусвободное. Рассматриваемое понятие не связано напрямую с понятием коммерческого программного обеспечения. Термин «проприетарное программное обеспечение» используется FSF (фондом свободного ПО) для определения программного обеспечения, которое с позиции Фонда не является свободным или полусвободным. Слова англ. *proprietary software* обозначают программное обеспечение, которое имеет собственника, осуществляющего контроль над этим программным обеспечением. Таким образом, этот термин может быть использован ко всему программному обеспечению, которое не находится в общественном использовании. Однако слово *proprietary* иногда используется в рекламе как

«владение монопольными правами на что-нибудь». Так и Фонд свободного программного обеспечения использует термин для выделения того, что собственник является основным фактором, в контрасте со свободным ПО, где этим фактором является свобода компьютерных пользователей.

Проактивные технологии — совокупность технологий и методов, используемых в антивирусном программном обеспечении, основной целью которых, в отличие от реактивных (сигнатурных) технологий, является предотвращение заражения системы пользователя, а не поиск уже известного вредоносного программного обеспечения в системе. При этом проактивная защита старается блокировать потенциально опасную активность программы только в том случае, если эта активность представляет реальную угрозу. Серьезный недостаток проактивной защиты — блокирование легитимных программ (ложные срабатывания).

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

IoT – Internet of Things

RFID – Radio Frequency IDentification

WSN – Wireless Sensor Network

PLC – Power Line Communication

IPv6 – Internet Protocol version 6

6LoWPAN – IPv6 over Low power Wireless Personal Area Networks

SOA – Service Oriented Architecture

ПО – программное обеспечение

URI – Unified Resource Name

HTTP – Hypertext Transfer Protocol

IP – Internet Protocol

TCP – Transmission Control Protocol

CoAP – Constrained Application Protocol

REST – Representational State Transfer

API – Application Programming Interface

LLNs – Low-power and lossy networks

CSV – Comma Separated Values

DoS – Denial of Service

M2M – Machine-to-machine

ЭО ПАП – экспериментальный образец масштабируемой сервис-ориентированной программно-аппаратной платформы

VDS – виртуальный выделенный сервер

VE – виртуальная среда

VPS – виртуальный частный сервер

ПАС – программно-аппаратный стенд

СУБД – система управления базами данных

## ВВЕДЕНИЕ

В данном промежуточном отчете по 1-му этапу ПНИ «Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)» приведены результаты и обобщения теоретических исследований поставленных перед ПНИ в целом.

Общей целью работы является разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы использующей современные технологии беспроводных сенсорных и агентных сетей, технологий семантического веба и технологии облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств. А также целью ПНИ является внедрение полученных результатов в проектах, реализуемых совместно с Индустриальным партнером и связанных с разработкой: умных сетей тепло-, водо- и электроснабжения, программной платформы для реализации современной городской среды на основе повсеместного использования информационно-коммуникационных технологий, и универсальной программной платформы для автоматизированного управления зданиями.

Актуальность работы обусловлена постановкой новой научно-технической задачи обеспечения синтаксической и семантической интероперабельности компонентов распределенной сети электронных потребительских устройств, применением новых методов и инструментов технологий семантического веба.

Среди конкретных научно-исследовательских, научно-технических областей, подлежащих детальной проработке в данной ПНИ, следует выделить следующие области:

- а) Математические методы и алгоритмы агрегации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных с применением приемов онтологического инжиниринга и технологий семантического веба.
- б) Математические методы и алгоритмы нормализации и анализа больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных генерируемых в распределенной сети электронных потребительских устройств (Internet of Things).
- в) Математические методы и алгоритмы интерактивной визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных на основе технологии связанных данных (Linked Data).
- г) Технические принципы и методические подходы к организации и развертыванию решений по агрегации, нормализации, анализу и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных беспроводных сенсорных сетей, осуществляющих сбор, первичную обработку, семантическое аннотирование и передачу данных от потребительских устройств к центрам обработки данных.
- д) Методические подходы к передаче данных от потребительских устройств к центрам обработки данных (далее – ЦОД).
- е) Технические принципы и методические подходы к организации защиты информации, передаваемой в распределенной сети электронных потребительских устройств (Internet of Things).

# **1   Аналитический   обзор   современной   научно-технической,   нормативной,   методической   литературы, затрагивающей научно-техническую проблему, исследуемую в рамках ПНИ, в том числе обзор научных информационных источников**

## **1.1   Предпосылки созданию распределенной сети электронных потребительских устройств (Internet of Things)**

Удешевление стоимости и ускорение производства электронных потребительских устройств уже привело к внедрению огромного количества этих устройств во все виды производства и жизнедеятельности человека. А появление новых протоколов идентификации и передачи данных, в том числе и для электронных устройств с низким энергопотреблением, таких как IPv6, RFID, 6LoWPAN, ZigBee и т.д., представило возможным подключение этих устройств к сети Интернет и реализацию межмашинного взаимодействия. Эти факторы привели к появлению в 1999 году концепции распределенной сети электронных потребительских устройств (далее – Интернет вещей) предложенной Кевином Аштоном [1].

К данному моменту присутствует растущий интерес к использованию технологий Интернета вещей в различных областях, таких как сельское хозяйство, пищевая и перерабатывающая промышленность, экологический мониторинг, системы безопасности и видеонаблюдения и многие другие. Например, использование технологий Интернета вещей в разработке интеллектуальной транспортной системы позволяет городским государственным органам получать информацию о скоплении и передвижении транспорта, а в итоге прогнозировать возможное местоположение транспортных средств и наличие загруженности на дорогах. Технологии Интернета вещей стали основой для создания новой научно-технической продукции в таких областях, как умные сети электроснабжения (Smart

Grid), интеллектуальные транспортные системы (Smart Transportation), системы мониторинга и прогнозирования состояния здоровья пациентов (Smart Health) и другие, которые решают определенные общественные потребности.

Согласно отчетам, полученных разными организациями, такими как Oracle (<http://www.oracle.com>), HP (<http://www.hp.com>), Intel (<http://www.intel.com>) и другие, к 2020 году количество устройств в распределенной сети электронных потребительских устройства будет составлять около 50 миллиардов единиц.

Согласно утверждению Кевина Аштона, за прошедшие годы после предложения разработки распределенной сети потребительских устройств, уже произошли существенные изменения. Используемые RFID-устройства и другие датчики получают данные о мире, позволяя наблюдать, устанавливать связи и понимать мир без ограничений, накладываемых на информацию, получаемую от человека. Это, по его мнению, не просто «обновленные штрих-коды» или «новый способ управления скоростными магистралями», а ключевая новая система, которая приведет к существенным изменениям в обществе подобно таким, какие произошли после появления интернета – или даже большим [1].

## 1.2 Определение распределенной сети электронных потребительских устройств (Internet of Things)

Atzori и другие в статье [2] утверждают, что Интернет вещей состоит из трех образующих парадигм (см. рисунок 1.1): интернет-ориентированная, образуемая связующим программным обеспечением; сенсор-ориентированная, образуемая распределенной сеть электронных потребительских устройств; семантически-ориентированная, образуемая данными и знаниями о предметной области. Хотя такое разделение объясняется междисциплинарной природой Интернета вещей, но реальная полезность Интернета вещей в какой-либо области может быть достигнута только при наличии пересечения всех трех предложенных парадигм.

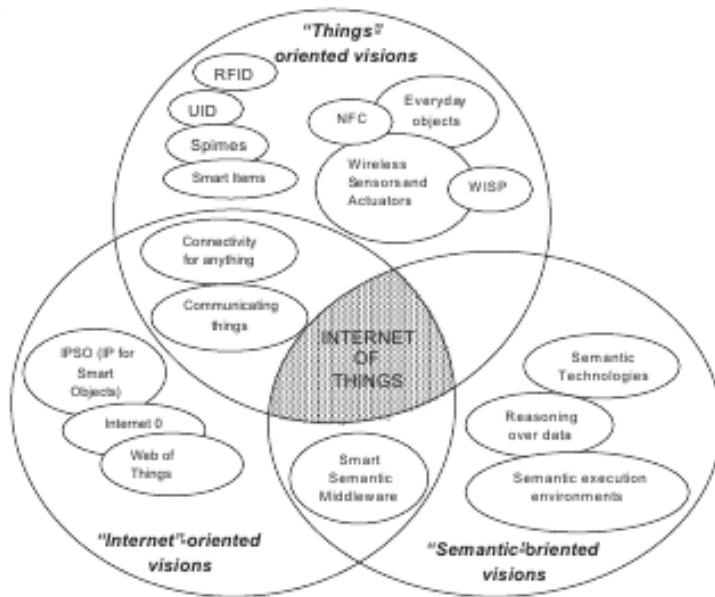


Рисунок 1.1 - Internet of Things как комбинация трех образующих его парадигм

Рабочая группа под названием RFID Special Interest Group (<http://www.ifla.org/rfid>) определяет Интернет вещей как:

«Всемирная сеть взаимосвязанных объектов имеющих уникальную схему идентификации на основе стандартных протоколов передачи данных».

Согласно определению, предложенным Кластером европейских исследовательских проектов (CERP IoT) [3], Интернетом вещей являются: «”Вещи”,

являющимися активными участниками бизнес, информационных и социальных процессов, где они взаимодействуют друг с другом и окружающей средой, через обмен данными и информацией об окружающей среде, реагируя в автоматическом режиме на события реального/физического мира и производя новые события в рамках какой-либо процесса с и без участия человека», в соответствии с рисунком 1.2.

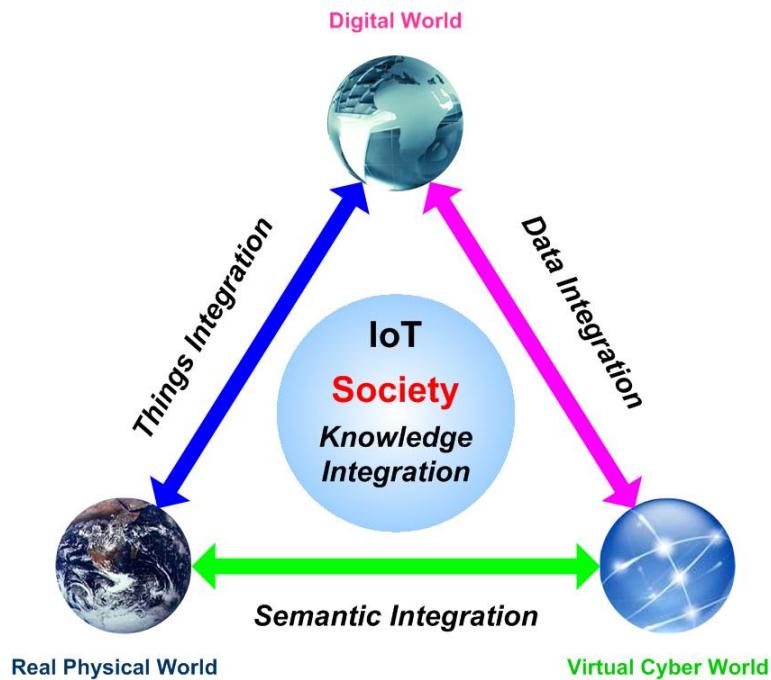


Рисунок 1.2 - Internet of Things – взаимодействие физического, цифрового и виртуального миров и общества

Согласно компании Forester Research [4]: «Интернет вещей использует информационные и коммуникационные технологии для, чтобы сделать критические инфраструктурные компоненты и сервисы, относящиеся к городской администрации, образованию, здравоохранению, правоохранительным органам, агентства недвижимости, транспорту и обслуживающим организациям, более интерактивными, эффективными и подготовленными», в соответствии с рисунком 1.3.

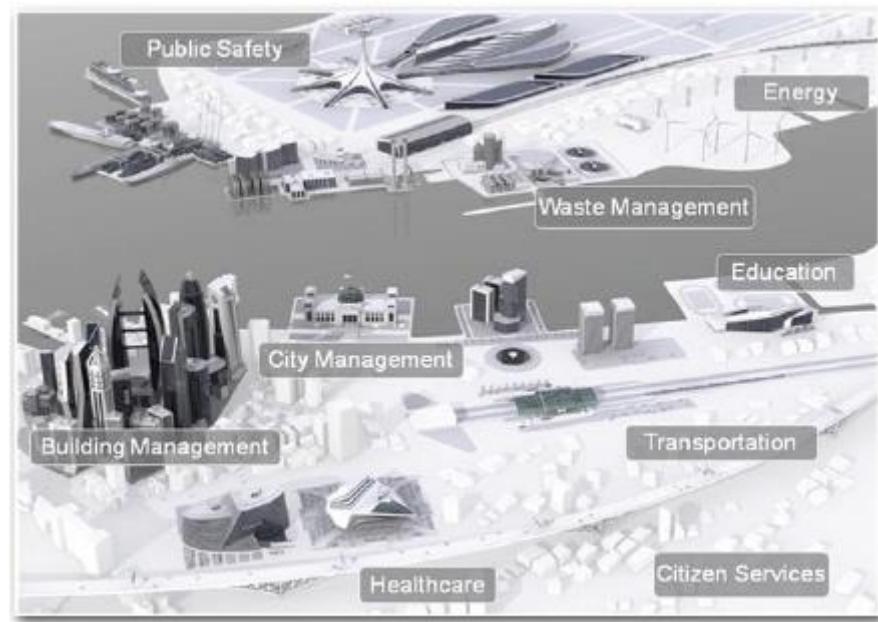


Рисунок 1.3 – Области городской администрации, позволяющие использовать Internet of Things для увеличения эффективности управления и обслуживания

Определение Интернета вещей предложенное J. Gubbi [5]: «Взаимодействие сенсорных и агентных устройств, предоставляющих возможность обмениваться информацией между информационными платформами через унифицированный интерфейс, тем самым развивая общую картину для инновационных приложений. Это достигается через интеллектуальный сбор данных, анализ данных и представление информации, опираясь на облачные технологии и платформы, как на унифицированный интерфейс», в соответствии с рисунком 1.4.

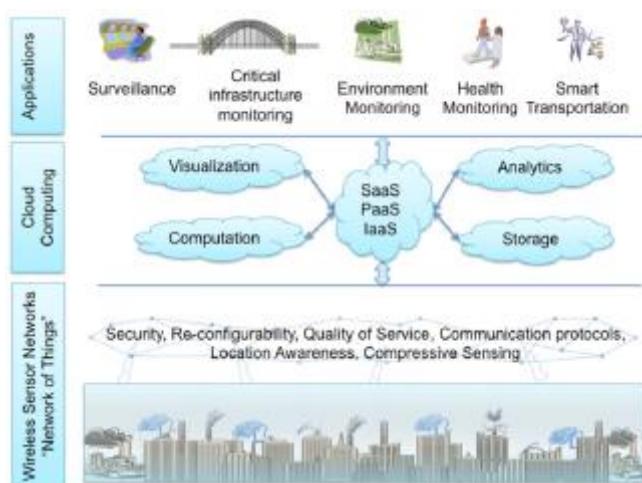


Рисунок 1.4 – Концепция IoT во главе с облачными технологиями

## **1.3 Элементы технологий Интернета вещей**

### **1.3.1 Краткий обзор элементов технологий Интернета вещей**

В этом разделе перечисляются категории элементов, образующих технологии Интернета вещей. Более подробное описание некоторых подкомпонентов в статьях. Существуют три категории элементов, составляющих Интернет вещей:

- a) аппаратное обеспечение, включающее сенсоры, механизмы управления и встраиваемые передатчики;
- б) программное обеспечение, включающее хранилища данных и инструменты анализа данных;
- в) подходы к представлению данных, включающие новые инструменты для визуализации и интерпретации данных, которые будут широкодоступные на различных платформах и спроектированы для различных областей применения.

Далее рассматриваются несколько технологий из перечисленных категорий, которые составляют Интернет вещей.

### **1.3.2 Технология радиочастотной идентификации (RFID)**

Технология радиочастотной идентификации (RFID) – значительный прорыв области встроенных технологий коммуникации, которые позволяют создавать микрочипы для беспроводной передачи данных. Технология позволяет автоматически идентифицировать любой предмет, к которому прикреплён датчик (также называемый RFID-тэгом), работающий по технологии RFID, что является неким электронный штрих кодом [6, 7]. На рисунке 1.5 представлены фотографии основных компонентов экосистемы RFID.



Рисунок 1.5 - Экосистема RFID. Приемные антенны (слева), метки (справа внизу) и терминал (справа вверху), позволяющий связывать метки с вещами

Пассивные RFID-тэги не имеют своего источника питания, поэтому они используют энергию, передаваемую устройством,читывающим данные с RFID-тэгов, чтобы передать необходимую информацию. Технология применяется в розничных продажах и при контроле грузоперевозок, а также на транспорте, где бумажные билеты заменяются на пластиковые карты со встроенным RFID-тэгом, и в системах контроля доступа. Кроме того пассивные RFID-тэги используются в банковских картах и в картах оплаты проезда на платных дорогах. Активные RFID-тэги имеют собственный источник питания и могут сами инициализировать обмен данными. Среди нескольких областей применения выделяется, система мониторинга грузов в порту [7].

### 1.3.3 Технологии беспроводных сенсорных сетей (WSN)

Недавний технологический прогресс в развитии интегральных схем с низким энергопотреблением и технологий беспроводной передачи данных привел к появлению эффективных, дешевых устройств небольших размеров и низким энергопотреблением, которые используются в системах, где необходимо удаленно считывать какие-либо показатели. Эти факторы повысили жизнеспособность распределенных сетей сенсоров, состоящих из большого количества интеллектуальных сенсоров, которые позволяют собирать, обрабатывать и

анализировать информацию в разных окружающих условиях [8]. На рисунке 1.6 проиллюстрированы узлы беспроводной сенсорной сети.

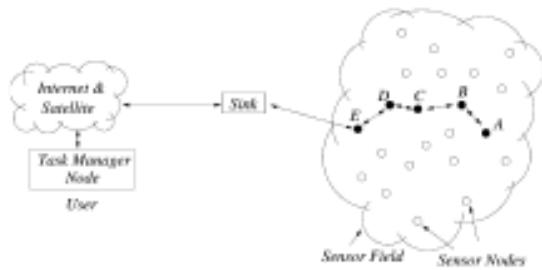


Рисунок 1.6 - Сенсорные узлы, взаимодействующие в сенсорном поле

Активные RDF-тэги схожи с узлами беспроводной сенсорной сети с маленьким объемом рабочей памяти и ограниченными вычислительными ресурсами. Исследовательские задачи, которые должны быть решены для реализации всего потенциала беспроводных сенсорных сетей, являются по своей природе междисциплинарными [8]. Данные с сенсоров передаются между сенсорами (узлами сети) и отправляются в центральную или распределенную систему для анализа. Беспроводная сенсорная сеть состоит из следующих компонентов:

- а) аппаратное обеспечение – как правило, узел состоит из сенсора, вычислительного модуля, модуля передачи данных и источника питания;
- б) стек протоколов передачи данных – узлы сети размещаются по-разному, в зависимости от области применения. Проектирование нужной топологии и маршрутизации является важной задачей, от которой зависит масштабируемость и срок жизни сети. Узлы беспроводной сенсорной сети передают данные на базовую станцию в односкачковом или многоскаковом режиме. Базовая станция, аккумулирующая данные, которые поступают со счетчиков, должна поддерживать необходимый стек протоколов передачи данных для того чтобы

обеспечить связь сети с внешним миром. Такая станция служит шлюзом, соединяющим беспроводную сенсорную сеть с Интернетом.

- в) программное обеспечение – ПО, которое реализует механизм объединения сервис-ориентированной архитектуры и беспроводных сенсорных сетей для предоставления доступа к гетерогенным сенсорным данным в не зависимости от способа развертывания сети [9]. Такой подход основывается на идеи изоляции ресурсов, которые могут быть использованы одновременно несколькими системами. Для этого требуется ПО, основанное на архитектуре Open Sensor Web Architecture (OSWA) [10]. OSWA включает в себя унифицированный набор операций и стандартных моделей данных, которые определены рабочей группой Open Geospatial Consortium (OGC).
- г) методы защищенного сбора данных – для увеличения времени работы сети и также улучшения стабильности передачи данных, собранных с сенсоров, необходимо использовать методы эффективной и защищенной агрегации данных [10]. Типичной задачей беспроводных сенсорных сетей является поддержание работы сети в случаях сбоя в работе одного или нескольких узлов. В тоже время обеспечение защищенности сети является необходимость, чтобы защититься от злоумышленников.

#### **1.3.4 Методы адресации электронных потребительских устройств**

Возможность однозначно адресации устройства по его уникальному идентификатору является решающим условием успеха Интернета вещей. Кроме адресации устройств, это позволит иметь возможность управлять миллиардами электронных потребительских устройств через Интернет. Способ адресации должен обладать следующими характеристиками: уникальность, надежность, постоянство и масштабируемость. Широко используемый на данный момент протокол IPv4 может быть использован для этого, но только с некоторыми ограничениями, который не позволяют использовать такой способ в Интернете. Протокол IPv6 решает существующую проблему его предшественника, увеличивая количество IP-адресов,

которые могут быть присвоены устройству, но различия в природе узлов в беспроводных сенсорных сетях, типах и моделях данных, методе выполнения параллельных операций не позволяет в полной мере решить проблему [11].

Долговременное и стабильное функционирование сети передачи данных является еще одним аспектом Интернета вещей. Хотя TCP/IP заботится об этом аспекте, перенаправляя трафик эффективно от источника до получателя, но узким местом остается «интерфейс» между внешней сетью и беспроводной сенсорной сетью. Более того, масштабируемость системы адресации устройствам должна быть устойчивой. Добавление новой подсети или одиночного устройства в общую сеть не должно сказываться на её производительности, работе устройств, надежности передачи данных или эффективном использовании устройств через пользовательские интерфейсы.

Решением этой задачи рассматривается схема единообразного именования ресурса (Unified Resource Name, URI). URN создает отпечаток ресурса, к которому можно получить доступ через URL (Unified Resource Location). Имея большое количество собираемой пространственной информации, очень важно использовать дополнительные метаданные для передачи информации из базы данных к пользователю по Интернету [12]. Другой важной разработкой является легковесный IPv6, что позволяет использовать его для адресации устройств с ограниченными вычислительными ресурсами.

### **1.3.5 Методы хранения и анализа данных**

Одним из последствий развития Интернета вещей является беспрецедентное количество вновь созданных данных. Методы хранения, владения и вычисления срока истечения хранения данных становятся ключевыми. Интернет суммарно потребляет около 5% всей генерируемой сегодня энергии и с ростом потребности, это число непременно вырастит. Следовательно, центры обработки данных, которые работают на генерируемой энергии, захотят обеспечить себя доступом к эффективным и надежным источникам энергии. Данные мониторинга и исполнения

должны храниться и использоваться разумно и эффективно. Важно разрабатывать алгоритмы искусственного интеллекта, которые могут работать как на одиночном сервере, так и на распределенном кластере серверов. Необходимо разрабатывать новые алгоритмы интеграции данных, которые позволили бы анализировать данные. Также необходимы алгоритмы автоматического принятия решения, которые могли бы базироваться на технологиях машинного обучения, генетических алгоритмах, нейронных сетях и других. Такие системы имеют такие характеристики как интероперабельность, интегрируемость и адаптивность. Они также имеют модульную архитектуру, как на аппаратном, так и на программном уровне, что хорошо подходит для Интернета вещей. Более того необходима централизованная инфраструктура для хранения и анализа данных. Технологии облачных хранилищ набирают популярность в последние годы. Тоже происходит с облачными платформами для анализа и визуализации данных.

### **1.3.6 Методы и технологии визуализации данных**

Визуализация данных является критической составляющей приложений Интернета вещей, так как это позволяет пользователю взаимодействовать с окружающей средой. С учетом последних достижений в разработке технологий создания сенсорных экранов, использование смарт-планшетов и телефонов стало более интуитивно. Для привлечения неспециалистов к использованию приложений Интернета вещей должны создаваться привлекательные и удобные пользовательские интерфейсы. По мере перехода от 2D к 3D-экрана, для потребителей информация может быть предоставлена в более удобном виде. Это также позволит чиновникам выделять какую-то полезную информацию из сырых данных, которая крайне необходима для быстрого принятия решений. Извлечение полезной информации из исходных данных является нетривиальной задачей. Это включает в себя обнаружение и визуализацию, связывание и моделированные данных, информации, представленной в соответствии с потребностями конечного пользователя.

## **1.4 Обзор тенденций в развитии технологий Интернета вещей**

Безусловно, основные плюсом развития и внедрения технологий Интернета вещей является влияние, которое они окажут на некоторые аспекты повседневной жизни и поведения потенциальных пользователей. С точки зрения конкретного пользователя, наиболее очевидные последствия развития Интернета вещей будут видны как на бытовом уровне, так и в работе. В этом контексте, системы частичного ухода, системы «умных домов и офисов», системы «электронного здравоохранения», системы электронного образования, это лишь несколько примеров возможного применения сценариев, в которых новая парадигма будет играть ведущую роль в ближайшем будущем [2]. Аналогично, с точки зрения хозяйственной деятельности, наиболее очевидными последствиями будут одинаково видны в таких областях, как автоматизация и промышленное производство, логистика, управление бизнес-процессами, интеллектуальные транспортные системы для перевозки людей и грузов.

Однако все еще существует много непростых задач как технологических так и социальных, которые требуют решения для того чтобы технологии Интернета вещей стали активно применяться в реальной жизни. Здесь важнейшими вопросами являются вопросы достижения полной интероперабельности между взаимосвязанными потребительскими устройствами, и как сделать их достаточно «умными», чтобы обеспечить их адаптивность и автономность, в тоже время, гарантируя доверие, безопасность и конфиденциальность их пользователей и их данных, в соответствии с рисунком 1.7. Кроме того, Интернет вещей вызовет появление нескольких новых проблем, касающихся вопросов эффективного использования ресурсов распределенной сети потребительских устройств.

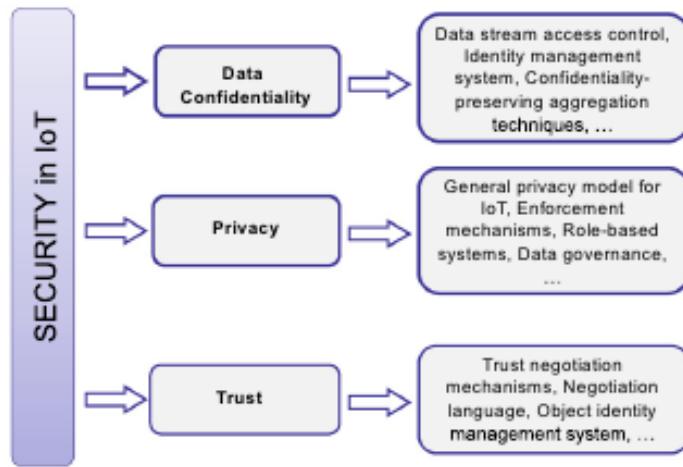


Рисунок 1.7 - Графическое представление вопросов безопасности I-o-T

Исследовательская и консалтинговая компания Gartner в своем отчете о «циклах зрелости технологий», в соответствии с рисунком 1.8, определяет Интернет вещей как один из развивающихся ИКТ технологий. «Цикл зрелости технологий» - это способ представить развитие, применение, зрелость и влияние определенных технологий. Было предсказано, что технологии Интернета вещей через 5-10 лет будут активно применяться во многих областях жизнедеятельности.

Популярность различных концепций и технологий изменяется со временем. Популярность запросов к поисковому сервису компании Google для терминов Internet of Things, Wireless Sensor Networks и Ubiquitous Computing за последние 10 лет представлена на рисунке 1.9. Как видно из этого рисунка, количество поисковых запросов для термина Internet of Things постоянно увеличивается, а количество запросов для термина Wireless Sensor Networks снижается.

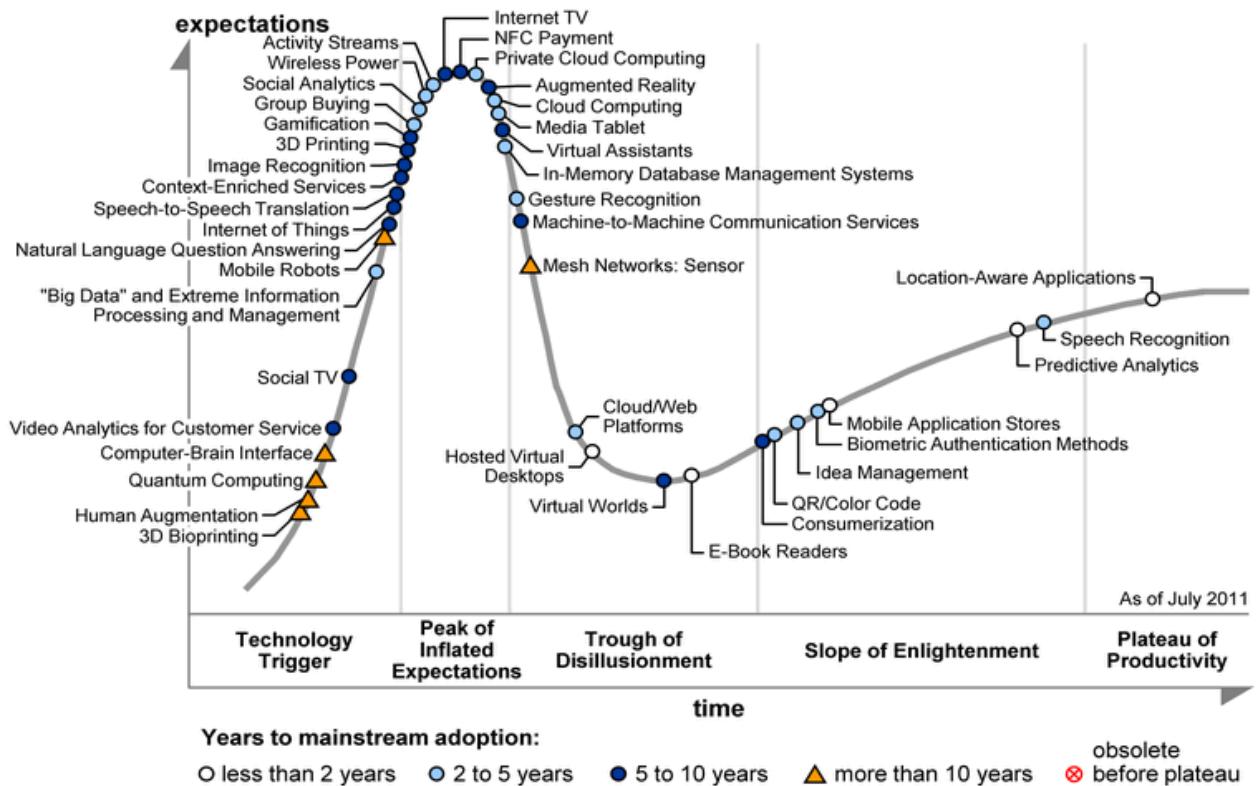


Рисунок 1.8 - "Цикл зрелости технологий" от Gartner

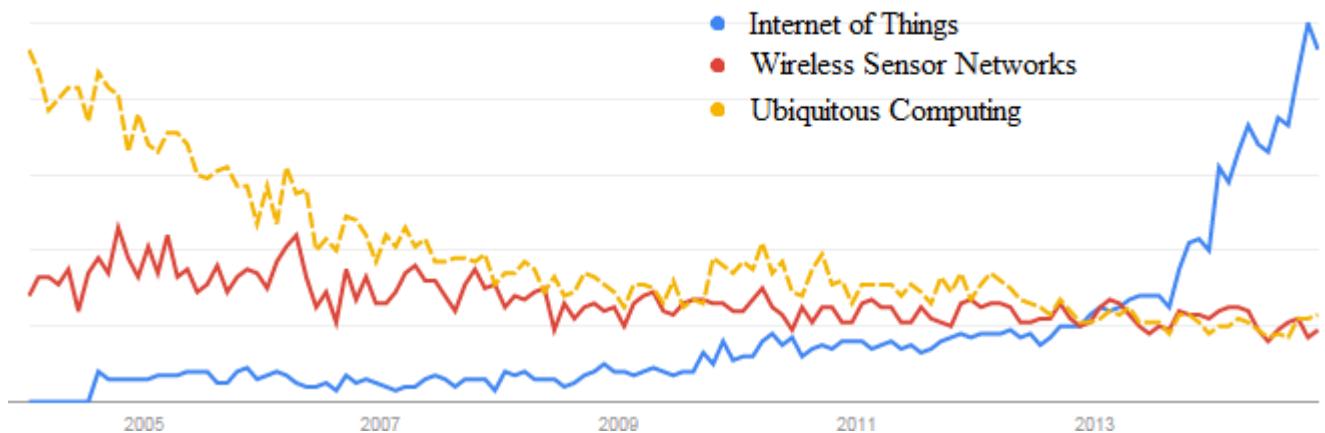


Рисунок 1.9 - Поисковые тренды Google с 2004 года для Internet of Things, Wireless Sensor Networks, Ubiquitous Computing

Многие прогнозы относительно развития IoT не были достаточно точны. В частности, было предсказано, что количество узлов IoT к 2007 году превысит число сотовых телефонов. Даже сейчас количество устройств не приблизилось к предсказанному. Причиной этому можно выделить текущее отсутствие надежных,

основанных на стандартах, и недорогих решений сразу от нескольких производителей.

Однако активно продвигается и уже принят ряд стандартов, способных изменить статус-кво. В частности, «интернет вещей» строится на основе стандарта IEEE 802.15.4e, использующего решетчатую топологию сети с синхронизацией времени, что позволяет передатчикам большую часть времени оставаться неактивными и экономить энергию. Маршрутизация в такой сети основана на стандарте 6TSCH (Time-Synchronized Channel Hopping) for IPv6, позволяющем узлам найти друг друга. Упомянутая ранее работающая open-source реализация на основе данного подхода – OpenWSN. Однако потребуется несколько лет для создания действующего стандарта.

В ноябре компания Intel объявила о создании подразделения, в чьи задачи входит продвижение новых технологий, направленных на подключение электронных устройств к сети передачи данных. Подразделение носит название «Internet of Things Solutions Group» (IoTSG). IoTSG направит свои усилия на большинство рынков встроенных устройств, включая промышленную автоматизацию, аэрокосмическую и автомобильные отрасли, а также потребительскую электронику.

В декабре 2014 года компания Intel объявила о создании платформы для Интернета вещей IoT (Internet of Things), которая поможет другим производителям электроники быстрее и проще выпускать новые "умные" устройства на рынок. Новая платформа призвана облегчить производителям процесс вывода устройств IoT из стадии пилотных проектов в готовые массовые решения, основанные на функциональных блоках, которые могут быть объединены в требуемые комбинации, и предоставляемом программном обеспечении. С целью демонстрации возможностей IoT американская компания по недвижимости Rudin Management, которая разработала свою собственную систему программного обеспечения под названием DiBoss, показала, как она грамотно организовала управление энергией и

другими системами в своих зданиях. За год компания сэкономила существенные внушительные суммы с каждого здания [13].

Развитие инфраструктуры Интернета вещей вероятнее всего будет идти в соответствии с инкрементальным путем развития, тем самым переходят от менее сложных задач к задачам требующих значительных временных и материальных ресурсов. Для решения выше описанных научно-технических проблем требуется международное взаимодействие и системных подход [17, 18, 19, 20]. Кроме научно-технических задач перечисленных выше, мы также определили несколько исследовательских трендов:

- а) Интеграция социальных сетей и технологий Интернета вещей,
- б) Разработка экологических («зеленых») технологий Интернета вещей,
- в) Разработка контекстно-зависимого программного обеспечения для Интернета вещей,
- г) Использование технологий искусственного интеллекта для разработки интеллектуальных электронных потребительских устройств,
- д) Тесная интеграция технологий облачных вычислений и Интернета вещей.

## **1.5 Обзор существующих исследовательских проблем**

### **1.5.1 Основные исследовательские проблемы**

Общепризнано, что технологии и приложения Интернета вещей все еще находятся на ранней стадии развития [18]. Существует большое количество исследовательских и технических проблем [2, 14], такие как программная и аппаратная архитектура, стандартизация, безопасность, конфиденциальность и т.д. Необходимо приложить достаточно усилий, чтобы решить эти проблемы и исследовать характеристики различных областей применения Интернета вещей, чтобы обеспечить наиболее приемлемое встраивание в окружения различных областей применения. В данном разделе перечисляются основные исследовательские и технические проблемы Интернета вещей.

### **1.5.2 Архитектура Интернета вещей**

Исследование различных подходов к архитектуре Интернета вещей необходимо для представления, организации и структурирования Интернета вещей для его эффективного функционирования. В частности распределенная и гетерогенная природа Интернета вещей требует применения аппаратного, сетевого и программного обеспечения, процессов способного поддерживать такие устройства, сервисы, которые они предоставляют, и рабочие процессы, которые они затрагивают.

За последнее время было предложено некоторое количество аппаратных и сетевых архитектур, которые поддерживают распределенные вычислительные среды необходимые для Интернета вещей. Эти архитектуры включают в себя пикировые [19] и автономные [20].

Программная архитектура необходима для предоставления доступа к электронным потребительским устройствам и обеспечения обмена сервисами, предоставляемыми ими же. В частности сервис-ориентированная архитектура [21, 22, 23] и REST-модель [24, 25, 26] чаще всего предлагаются в качестве программной архитектуры Интернета вещей, так как они сфокусированы на сервисы и гибкость.

Интернет вещей определенно имеет, и будет иметь влияние на бизнес-процессы предприятий. Архитектура процессов необходима для эффективного структурирования бизнес-процессов, в которые будет интегрироваться Интернет вещей. В частности, исследователи изучили [27, 28], как структурированные рабочие процессы поддерживают всепроникающие вычислительные среды.

На данный момент не консенсуса о какой-либо одной архитектуре, которая бы наилучшим образом подходила бы для Интернета вещей. В большом количестве литературы предлагаются различные концептуальные архитектурные подходы, пока другие предложили критерий для оценки предложенных архитектур [29], а также концептуальную архитектуру, которая отвечает требованиям «умных вещей» [30].

### **1.5.3 Адресация и присваивание имен**

Поиск эффективных подходов к извлечению информации генерируемой различными объектами является еще одной исследовательской проблемой. Так же эту проблему можно описать двумя вопросами: Как идентифицировать и взаимодействовать с огромным количеством объектов? И как мы можем выполнять это эффективно, если устройства или объекты по своей природе имеют гетерогенные характеристики?

Адресация это механизм, который используется для идентификации объектов (например, компьютеры, роутеры, небольшие устройства) в сети. В сетях, основанных на протоколе IP, существуют два варианта: IPv4 и IPv6. Однако так как IPv4 основывается на 32-х битных адресах и Internet Assigned Numbers Authority (IANA) выдало последний блок адресов IPv4 в Июле 2011, то IPv6 видится единственным решением. IPv6 предоставляет широкое адресное пространство, так как этот протокол основывается на 128-битных адресах, что определенно позволит адресовать все необходимые устройства. Однако, даже если вышеизложенное обнадеживает, но поиск единственного решения является серьезной задачей, так как разными технологиями используются разные длины адресов.

### **1.5.4 Автоматическое конфигурирование электронных потребительских устройств**

В традиционном программно-аппаратном обеспечении, к системе подключается только некоторое ограниченное количество сенсоров, например, умный дом. С другой стороны Интернет вещей подразумевает объединение миллиардов сенсоров в одну сеть. В результате чего мы получаем необходимость в создании подходов для автоматического подключения и конфигурирования сенсоров. Из-за масштаба, не реально выполнить ручное подключение сенсоров к приложениям или промежуточному ПО [31]. Для того чтобы достичь этого необходимо, чтобы приложения понимали сенсоры, т.е. возможности сенсоров, схему данных, которые они производят, детали конфигурирования и т.д. Недавние разработки, такие как Transducer Electronic Data Sheet (TEDS) [32], стандарты

разработанные в Open Geospatial Consortium , Sensor Markup Languages (SensorML) [33], онтологии для описание сенсоров и их показаний [34], и Sensor Device Definitions [35] показывают направления будущих исследований, которые пытаются решить проблему автоматического или полуавтоматического конфигурирования сенсоров.

### **1.5.5 Стандартизация**

Быстрый рост Интернета вещей затрудняет стандартизацию в этой области. Однако, стандартизация играет важную роль в развитии и распространении Интернета вещей. Цель стандартизации снизить барьеры для появления новых провайдеров и пользователей, улучшение интероперабельности различных приложений и систем и так же повышения конкуренции продуктов и сервисов. Для того, чтобы убедиться в том что устройства и системы из разных стран могут взаимодействовать друг с другом необходим приложить некоторые усилия по стандартизации [14]. Различные стандарты, разработанные для Интернета вещей, могут стать серьезным двигателем распространения и развития. Выделяются несколько отдельных областей требующих стандартизации, такие как интероперабельность, радио частоты, семантическая интероперабельность, безопасность и конфиденциальность [36, 37, 38].

### **1.5.6 Контекстно-зависимые вычисления**

Как только электронные потребительские устройства подключены, необходимы методы для автоматического определения значения данных производимых этими устройствами и контекст, в котором они получены. Существует множество типов контекста, которые могут быть использованы для обогащения данных с таких устройств. Однако, понимание данных и соответствующее их аннотирование в Интернете вещей, где области применения обширны, является достаточно сложной задачей. Недавние разработки в технологиях семантического веба [39, 34, 40] и связанных данных [41, 42] показывают направления исследовательских трендов. Технологии семантического веба являются популярным механизмом для описания знаний предметной области.

### **1.5.7 Методы поиска и отбора сенсоров**

Данная проблема является одной из наиболее сложных. Очевидно, что в Интернете вещей мы имеем доступ к множеству различных сенсоров, расположенных в разных местах и т.д. Предположим, что ученому изучающему окружающую среду необходимо измерить загрязнение окружающей среды в каком-либо городе. Здесь появляются две проблемы: Какие из известных сенсоров измеряют загрязнённость окружающей среды? [43] Если существуют несколько сенсоров измеряющих одни те же явления, то который из них выбрать? [44] Что ответить на первый вопрос необходимо использовать знания соответствующей предметной области. Ручной поиск и отбор сенсоров, который предоставляют необходимую информацию, является не приемлемым из-за природы Интернета вещей. Для того чтобы ответить на второй вопрос необходимо использовать методы определения качества и точности измерений. Такие методы должны помочь ранжировать сенсоры на основе различных показателей, таких как точность, релевантность, надежность, стоимость и полноценность данных. Похожи проблемы существуют в области разработки веб сервисов и были предложены различные решения [45, 46], которые могут быть применимы в Интернете вещей.

### **1.5.8 Безопасность, конфиденциальность и доверие**

Электронный потребительские устройства в Интернете вещей обычно являются беспроводными и могут быть расположены в общедоступных местах. Беспроводная передача данных в сегодняшнем Интернете, как правило, является более защищенной, что было достигнуто через использования различных методов шифрования. Поэтому шифрование так же рассматривается ключевым компонентом обеспечивающим защищенность передачи данных в Интернете вещей. Однако, существующие устройства имеют недостаточно вычислительных ресурсов для того чтобы использовать существующие методы шифрования. Для того чтобы решить данную проблему необходимо разработать эффективные и более энергоэффективные алгоритмы шифрования [47, 48, 49].

Кроме того методы аутентификации и авторизации является также важной составляющей обеспечения защищенности и безопасности Интернета вещей. Поэтому необходимы методы, которые обеспечат однозначную идентификацию, авторизацию и аутентификацию электронных потребительских устройств [50, 48].

## **1.6 Обзор областей применения технологий Интернета вещей**

### **1.6.1 Классификация уровней областей применения**

Существуют несколько областей применения, которые будут затронуты развитием технологий Интернета вещей. Эти области могут быть классифицированы: по типу канала передачи данных, по покрытию, по масштабности, по гетерогенности, по степени вовлеченности пользователя и т.д. [21]. Также области применения можно разделить на четыре уровня: (1) системы для персонального и домашнего использования, (2) системы для использования на уровне предприятия, (3) системы для использования на уровне района или города, (4) мобильные системы, которые могут использоваться на всех уровнях. Уровни применения технологий Интернета вещей представлены на рисунке 1.10. Все уровни могут «сильно» пересекаться друг с другом, например, показания по потреблению электроэнергии в доме могут быть доступны для компании поставщика, который в свою очередь может использовать эти данные для оптимизации генерации электроэнергии. Интернет предоставляет возможность обмена информацией между различными провайдерами услуг.

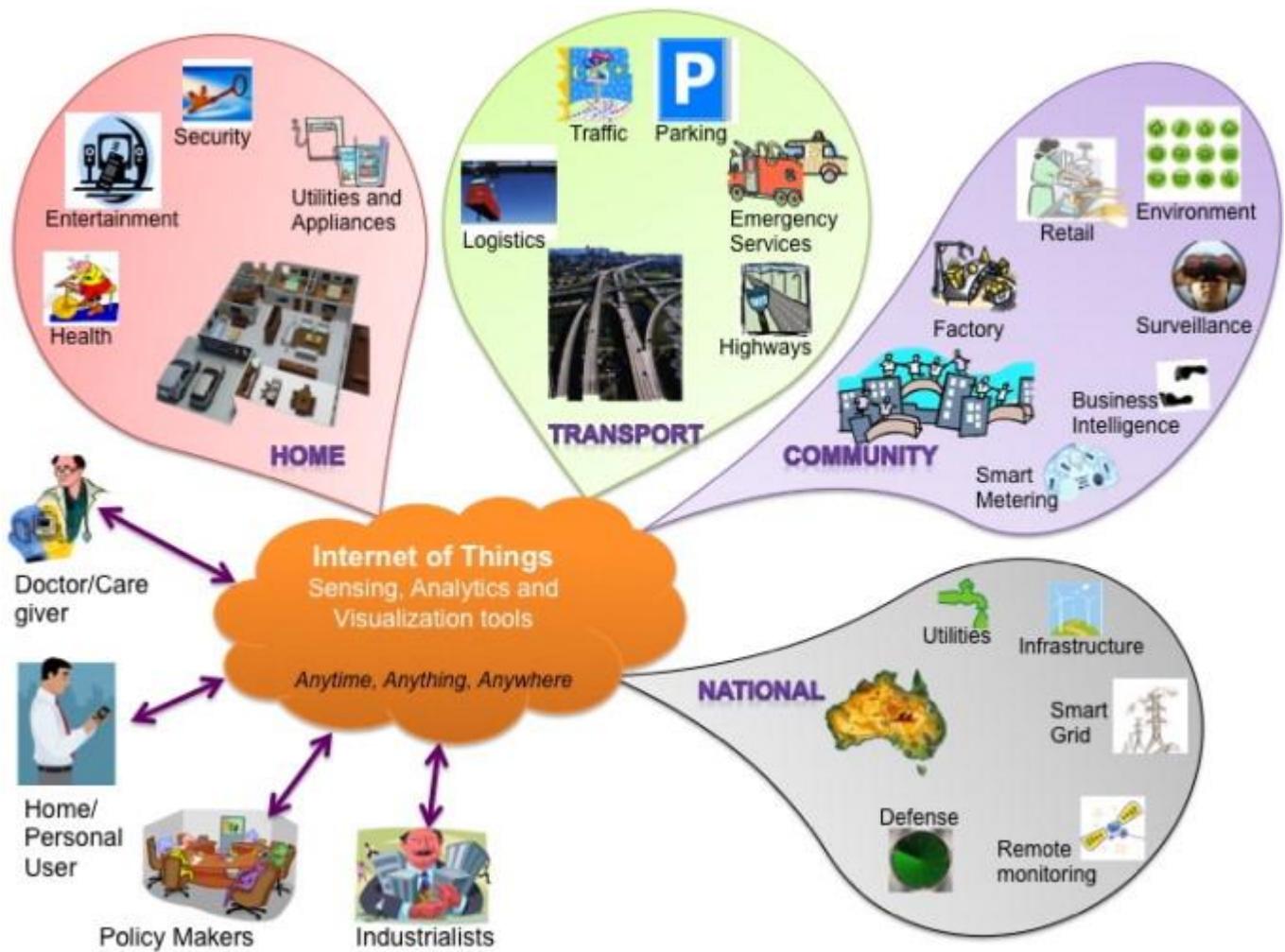


Рисунок 1.10 - Схематическое разделение областей применения технологий Интернета вещей

### 1.6.2 Области персонального и домашнего применения

Информация, собираемая из распределенной сети потребительских устройств, используется только теми, кому эта сеть принадлежит. Обычно для передачи данных в такой сети используется Wi-Fi как широкополосный канал передачи данных, в том числе видео и аудио.

Идею электронного здравоохранение (Ubiquitous healthcare) [2] предвидели в последние два десятилетия. Интернет вещей предоставляет отличную платформу для реализации этой идеи, где используются сенсоры, устанавливаемые на тело человека, и серверное обеспечение для предоставления доступа к показаниям с этих сенсоров. Например, смартфон может использоваться для сбора физических показателей через различные интерфейсы, такие как Bluetooth и другие. Пока

существуют несколько приложений под операционные системы Apple iOS, Google Android, Windows Phone, которые снимают различные физические измерения. Хотя они еще должны быть объединены через облачные технологии, чтобы предоставить врачам «универсальный» способ доступа к показаниям физического состояния пациента.

Развитие этого примера – это создания системы для мониторинга состояния престарелых людей в домашних условиях, которая позволяет врачу следить за своим пациентом, когда тот находится дома. Тем самым снижаются затраты на госпитализацию пациента, через ранее предупреждение заболевания или осложнений и лечение [52, 53].

Управление бытовыми приборами, такими как кондиционеры воздуха, холодильники, стиральные машинки и другие, позволит лучше управлять домом и потреблением энергии. Здесь « рядовые » пользователи становятся вовлеченными в парадигму Интернета вещей, так же как они вовлекались в парадигму Интернета [24, 25]. Социальные сети привносят в Интернет вещей свою сеть из миллиардов взаимосвязанных «вещей» [16, 26]. Интересным приложением было бы использование Твиттера, как «места», где каждое из устройств периодически публикует свои показания, которые могли бы быть прочитаны кем-либо откуда-либо. Хотя такой подход предоставляет универсальный инструмент для доступа к информации, но он потребует новых методов обеспечения безопасности и конфиденциальности [27].

### **1.6.3 Области применения на уровне предприятия**

Информация, собираемая в распределённой сети потребительских устройств, используется владельцем оборудования, но и частично предоставляется доступ к ней из внешних систем. Системы мониторинга окружающей среды - одна из наиболее распространённых областей применения Интернета вещей, которые используются для управления оборудованием в зданиях, например, освещение, отопление, вентиляция, кондиционирование и т.д.

Сенсоры довольно долго время являлись и являются неотъемлемой частью заводских установок для автоматизации, безопасности и климат контроля. Эта система, в конце концов, расширяется беспроводной сенсорной сетью, которая предоставляет гибкость в изменении установок каким-либо образом. Это ничто иное как «подсеть» Интернет вещей предназначенная для поддержки работы предприятия.

Одной из главных областей применения, которые уже привлекают все общее внимание, является Интернет вещей «умного окружения» (Smart Environment IoT) [52, 58]. Уже существует несколько реализованных испытательных лабораторий и еще больше планируются в ближайшие годы. Системы «умного окружения» включают подсистемы, перечисленные в таблице 1.1, так же в таблице кратко перечислены их технические характеристики.

Таблица 1.1 - Области применения систем "умного окружения"

Характеристики	Системы «умного дома/офиса»	Системы «умная розница»	Системы «умного города»	Системы «умного сельского хозяйства/лесоводства»	Системы «умная вода»	Системы «умный транспорт»
Размер сети	маленький	маленький	средний	средний/ большой	большой	большой
Пользователи	мало, члены семьи	немного, сообщество	много, чиновники, население	немного, владельцы земель, чиновники	немного, госслужащие	много, население
Энергия	перезаряжаемые батареи	перезаряжаемые батареи	перезаряжаемые батареи, аккумуляторной энергии	аккумулирование энергии	аккумулирование энергии	перезаряжаемые батареи, аккумулирование энергии
Способ передачи данных	Wi-Fi, 3G, 4G LTE	Wi-Fi, 3G, 4G LTE	Wi-Fi, 3G, 4G LTE	Wi-Fi, спутниковая связь	спутниковая и микроволновая связь	Wi-Fi, спутниковая связь

## Продолжение таблицы 1.1

Управление данными	локальный сервер	локальный сервер	коллективный сервер	локальный и/или коллективный сервер	коллективный сервер	коллективный сервер
Характеристики	Системы «умного дома/офиса»	Системы «умная розница»	Системы «умного города»	Системы «умного сельского хозяйства/лесоводства»	Системы «умная вода»	Системы «умный транспорт»
Типы устройств	RFID, WSN	RFID, WSN	RFID, WSN	WSN	одиночные сенсоры	одиночные сенсоры
Требования по пропускной способности	маленькие	маленькие	большие	средние	средние	средние/большие
Примеры	Aware home [28]	SAP future retail center [60]	Smart Santander [30], CitySense [62]	SiSViA [32]	GBROOS [33], SEMAT [34]	исследов. реализаций [35, 36]

### 1.6.4 Области применения на уровне района или города

Информация, собираемая с распределенной сети потребительских устройств в данной области применения, как правило, используется для оптимизации предоставления услуг, чем для предоставления каких-либо услуг « рядовым » пользователям. Например, компаниями поставщиками электроэнергии достаточно продолжительное время уже используются «умные счетчики» для автоматического удаленного сбора показаний энергопотребления для оптимизации работы энергетической сети. В данной области используются разные способы передачи данных: Wi-Fi, мобильный интернет, связь через линии передач (Power line communication, PLC) и другие.

Умные сети электроснабжения (Smart Grid) и системы интеллектуального учета энергопотребления (Smart metering) являются еще одними потенциальными областями применения технологий Интернета вещей, которые активно реализуются по всему миру [68]. Эффективное потребление энергии (электричества, тепла и т.д.)

может быть достигнуто постоянным мониторингом энергопотребления в доме и использованием этой информации для того чтобы управлять её потреблением. Например, такая информация может быть использована для того чтобы распределять нагрузку на энергосеть и тем самым поддерживать качество энергоресурса.

Интернет вещей, основанный на видеинформации [38], в котором интегрируются обработка изображений, компьютерное зрение и технологии передачи данных, поможет развивать исследовательские задачи на пересечении технологий обработки аудио и видео потоков, технологий инфракрасного излучения и технологий передачи данных. Системы наблюдения – одна из основных областей применения сетевых видеокамер, которые помогают отслеживать подозреваемых и подозрительную активность, находить безнадзорные предметы, обнаруживать попытки несанкционированного доступа.

Мониторинга воды в водоёмах и качества питьевой воды еще одна из критических областей, где используются технологии Интернета вещей. Сенсоры, снимающие показания критических параметров воды и устанавливаемые в необходимых точках, обеспечивают контроль качества поставки питьевой воды. Такие же системы могут использоваться для контроля орошения сельскохозяйственных угодий, а также и для контроля состояния почвы, что позволяет принимать обоснованные решения в вопросах сельского хозяйства [39].

### **1.6.5 Области широкого применения**

Интеллектуальные транспортные системы и логистические системы выделяются в отдельную категорию, из-за природы обмена информацией внутри самой системы и с внешними системами. Городской транспорт отрицательно влияет на уровни шумового загрязнения, загрязнения воздуха и выбросов углекислого газа. Пробки на дорогах в большинстве городов влияют на экономику и социальную сферу. Так же они влияют на грузоперевозки, вызывая задержки и отмены поставок. Динамический анализ транспортной информации повлияет на грузоперевозки,

улучшит планирование и прогнозирование. Интернет вещей позволит использовать широкомасштабное развертывание беспроводных сенсорных сетей для мониторинга в «режиме реального времени» времени затраченного в пути, маршруты передвижения, размеры транспортных пробок, загрязнение воздуха и шумовое загрязнение. Эти технологии также подготовят фундамент для разработки сценарных моделей при проектировании и планировании строительства или ремонта дорог, а также для улучшения систем контроля движения транспорта. Объединение информации с систем контролем движения транспорта и информации о транспортном состоянии дорог может быть использовано для предоставления новых услуг жителям городов [40].

Распространение технологии передачи данных Bluetooth среди различных устройств отражает распространение Интернета вещей в количестве устройств, таких как смартфоны, навигационные системы, автомобильные системы и т.д. Устройства, поддерживающие технологию Bluetooth, распространяют сигналы с уникальным идентификатором, называемым MAC-ID (Media Access Identification), которые могут быть получены сенсорами в радиусе доступа. Такие сенсоры, называемые считывателями, могут размещаться в различных местах для того чтобы обнаруживать передвижения устройств. Комбинирование этой информации с информацией из других источников, таких как светофоры, местоположение общественного транспорта, позволит решить различные задачи, такие как вычисления времени прибытия общественного транспорта, время нахождения на транспорте на автострадах и магистралях и т.д. [36]. Существуют несколько проблем с обеспечением безопасности в таких случаях, поэтому эта область активно развивается [41].

## **2 Выбор направления исследований в области разработки масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений**

### **2.1 Основные направления исследований**

В рамках этой главы отчета будут рассмотрены направления исследований, которые будут являться основными направлениями работ в рамках этого проекта. Рассматриваются следующие направления исследований:

- а) Методы и алгоритмы агрегации больших массивов гетерогенных данных с применением приемов онтологического инжиниринга и технологий семантического веба,
- б) Методы и алгоритмы нормализации и анализа больших массивов гетерогенных данных генерируемых распределенной сетью электронных потребительских устройств (Internet of Things),
- в) Методы и алгоритмы интерактивной визуализации больших массивов гетерогенных данных на основе технологий связанных данных (Linked Data),
- г) Подходы к передачи данных от электронных потребительских устройств к центрам обработки данных,
- д) Технические принципы и подходы к организации защиты информации, передаваемой в распределенной сети электронных потребительских устройств.

## **2.2 Методы и алгоритмы агрегации больших массивов гетерогенных данных с применением приемов онтологического инжиниринга и технологий семантического веба**

### **2.2.1 Основные протоколы агрегации данных Интернета вещей**

Интернет вещей быстро входит в нашу жизнь благодаря крошечным встроенным устройствам, которые напрямую подключены к сети Интернет по протоколу IP. Организация IETF (Internet Engineering Task Force) стандартизовала и продолжает стандартизовать несколько протоколов передачи данных для реализации на электронных устройствах с ограниченными ресурсами, такими как вычислительная мощность, размер оперативной памяти, объем источника питания, пропускная способность и т.д. Протокол 6LoWPAN [73] разработан для адаптации протокола IPv6 для маломощных сетей передачи данных с потерями, так же протокол RPL [74] для маршрутизации и сжатия данных для оптимизации протоколов следующего уровня. Вдобавок к этому стеку протоколов, развивается прикладной протокол передачи данных CoAP (Constrained Application Protocol), который объединяет этот стек в полноценный набор стандартизованных протоколов передачи данных. Такой набор протоколов делает технологии Интернета вещей интересными для промышленных систем: промышленные консорциумы ETSI M2M и IPSO Alliance адаптировали этот стек и разработали легковесный REST-протокол для обеспечения интероперабельности электронных потребительских устройств.

Когда идея Интернета вещей появилась, «вещи» были взаимосвязаны через их виртуальные представления с использованием штрих-кодов и RFID-меток. Позже были разработаны программные коммуникационные шлюзы для передачи данных между различными встраиваемыми системами. Далее появились реализации протокола IP [44, 45] для маломощных устройств, что привело к реализации прямого доступа по протоколу IP. Работы по стандартизации протоколов передачи данных проводимые IETF были сфокусированы только на протоколах транспортного уровня. Поэтому в различных проектах разрабатывались свои протоколы

прикладного уровня на основе протокола UDP, в результате чего устройства использующие протокол 6LoWPAN не могли взаимодействовать друг с другом.

Заметным кандидатом для изменения этой ситуации является протокол HTTP. HTTP протокол – это масштабируемый, устойчивый и используемый повсюду протокол, который де facto является прикладным протоколом сети Интернет. Однако производительность протокола TCP в сетях LLNs до пяти раз хуже чем протокола UDP [45]. Также для протокола TCP требуется относительно большое количество памяти для каждого соединения. Поэтому реализации протокола 6LoWPAN в комбинации с протоколом HTTP не очень производительны. Существуют реализации протокола HTTP по верх протокола UDP, такие как HTTPU, известный из UPnP [77], или EBHTTP [78], но не один из них не получил широкого применения. Как только появятся реализации стека протоколов TCP/IP для встраиваемых устройств, появится возможность применить кросс уровневую оптимизация для компенсации низкой производительности протокола HTTP для маломощных сетей. В проекте Smews [48], например, была достигнута разумная производительность протокола HTTP до 256 одновременных подключений.

Другим кандидатом является протокол CoAP, протокол передачи данных прикладного уровня похожий на HTTP, но специально спроектированный для встраиваемых узлов и сетей. В отличие от предыдущих попыток, CoAP не пытается сжимать протокол HTTP, а является совершенно новым протоколом, архитектура которого основана на принципах REST [49]. Это означает, что он имеет несколько ограничений по сравнению с протоколом HTTP, но в тоже время он не наследует все проектные решения протокола HTTP, поэтому он может лучше отвечать требованиям встраиваемых устройств. В частности, CoAP также превышает возможности HTTP в отношении межмашинного взаимодействия, например, он реализует поддержку push-уведомлений и IP multicast.

Имея два REST-протокола, которые покрывают полный спектр возможных электронных устройств, IPSO альянс инициировал следующих шаг к обеспечению

интероперабельности устройств в коммерческих продуктах. Так называемый «IPSO Profile» определяет стандартную структуру ресурсов встроенных веб-серверов. Он использует стандарты Web Linking [81] и CoRE Link Format [82] для определения типов ресурсов с предопределенными функциями и типами данных. На данный момент этот профиль позволяет описать устройства и информацию об их местоположении (например, производитель, уровень батареи, координаты и т.д.), формат сообщений об изменении статуса и предупреждениях, формат сообщений общего назначения, управление мощностью и светом, и сенсоры общего назначения. Для последнего, профиль использует спецификацию Unified Code for Units of Measure (UCUM), которая описывает уникальные идентификаторы для обозначения физических единиц измерения (например, W для вольт или Cel для градусов цельсия), включая префиксы, например, M для обозначения мега.

### **2.2.2 Протокол передачи данных прикладного уровня СоAP (Constrained Application Protocol)**

Протокол СоAP – это результат работы рабочей группы Constrained RESTful Environments (CORE) в организации IETF. Это протокол передачи данных прикладного уровня, цель которого реализовать возможность взаимодействия по принципам REST, как это реализовано в протоколе HTTP, но для устройств и в сетях с низкой пропускной способностью и ограниченными вычислительными ресурсами.

СоAP состоит из двух под-уровней: уровень сообщений и уровень запросов/ответов. Уровень сообщений расширяет протокол UDP с поддержкой механизма определения дубликатов и, если необходимо, механизма гарантированной доставки сообщений, реализованный через простую ретрансляцию с экспоненциальной задержкой. Уровень запросов/ответов позволяет использовать REST-подобное взаимодействие через унифицированные интерфейсы, адресуемые по URI, широко известные методы, такие как GET, PUT, POST и DELETE, и передачу информации о самом устройстве через адресуемые ресурсы. Что бы более «легковесным» чем протокол HTTP, СоAP поддерживает только ограниченный

набор функций HTTP и использует компактное, бинарное кодирование, которое было спроектировано для сериализации и разбора на «маленьких» устройствах.

В добавок к выше перечисленной функциональности, CoAP предоставляет следующие возможности:

- а) уведомления об изменение состояния ресурсов: CoAP позволяет клиентам получать уведомления об изменениях состояния через простой механизм публикации/подписки [83].
- б) блоковая передача данных: когда представления ресурсов становятся слишком большими для того чтобы быть переданными в рамках одной датаграммы, CoAP предоставляет механизм их передачи в виде блоков [84]. Этот механизм, например, позволяет загружать обновления прошивки без переключения на альтернативный протокол для передачи больших пакетов данных на или с устройства. Блоковая передача так же позволяет передавать большие пакеты данных «на лету» с помощью инкрементного подхода, что необходимо для устройств, которые могут обрабатывать только ограниченный объем информации в момент времени.
- в) коммуникация на уровне группы: CoAP так же предназначен для коммуникации внутри группы на основе механизма IP multicast. Однако не все аспекты этого механизма еще были описаны [85].
- г) обнаружение ресурсов: для обнаружения и идентификации ресурсов, которые предоставляет сервер, CoAP использует специализированный URI-путь /.well-known/core в соответствии со стандартом RFC 5785 [86]. Такие ресурсы описаны в стандарте CoRE Link Format [82], который основан на стандарте Web Linking [81]. Он определяет дополнительные атрибуты ссылок для семантических типов ресурсов («rt»), использования интерфейса («if»), формат данных («ct») и максимальный размер передаваемого пакета («sz»).

### 2.2.3 Обзор реализаций протокола СоAP

В данном разделе приводятся результаты опроса [56] проведенного между мартом и апрелем 2012 года на мероприятии организованном институтом European Telecommunications Standards Institute (ETSI) в Париже, Франция. В таблице 2.1 представлены результаты проведенного опроса, где участников попросили указать поддерживаемые языки программирования, платформы и целевые области применения их реализации протокола СоAP.

Таблица 2.1 - Реализации протокола СоAP

Имя компании или реализации	Лицензия	Язык програм.	Платформа	Описание
Conzorzio Ferrara Ricerche	-	NesC/C	TinyOS	полная реализация для устройств 1-го класса
ETH Zurich “Californium”	BSD	Java	JVM	полная реализация протокола
ETH Zurich “Copper”	BSD	JavaScript	Firefox	инструмент для тестирования
ETH Zurich “Erbium”	BSD	C	Contiki	для устройств 1-го класса
Hitachi	-	C	Embedded Linux	только серверная реализация протокола для устройств 2-го класса
IBBT	-	C++	Click modular Router	полная реализация для любых устройств
Intecs	коммерческая	C++	POSIX	для серверных приложений и встраиваемых шлюзов
KoanLogic “evcoap”	BSD	C	Linux	полная реализация общего назначения
NXP	-	Visual C	Windows XP	реализация для шлюзов к JenNet-IP устройствам
Patavina Technologies	коммерческая	C++	proprietary OS	для проводных и беспроводных устройств
Sensinode “Nanoservice Device Library”	коммерческая	C	-	кроссплатформенная реализация для устройств 1-го и 2-го классов

## Продолжение таблицы 2.1

Имя компании или реализации	Лицензия	Язык програм.	Платформа	Описание
Sensinode “Nanoservice Device Library”	коммерческая	Java	JVM	полная реализация для любых устройств
Университет Бремена “libcoap”	GPLv2 и BSD	C	POSIX и Contiki	общего назначения для устройств 1-го и 2-го классов
Университет Бремена “CoapBlip”	BSD	C	TinyOS	устройства 1-го класса
Имя компании или реализации	Лицензия	Язык програм.	Платформа	Описание
Университет Бремена “Bonsai”	-	C#	.NET	полная реализация для любых устройств
Университет Бремена “coap.me”	-	Ruby	-	для серверных приложений
Университет Ростока “jCoAP”	Apache 2.0	Java	JVM	полная реализация; для любых платформ
Watteco	-	C	Contiki	для устройств 1-го класса; основан на “Erbium”
-	-	C	Unix	реализация только клиентской части протокола для встраиваемых устройств
-	-	Java	JVM	реализация протокола для мобильных и встраиваемых устройств

Цель протокола CoAP предоставить возможность взаимодействия с «электронными устройствами с ограниченными ресурсами» в соответствии с архитектурными принципами REST. Чтобы лучше описать такие устройства рабочая группа Light-weight implementation Guidance (IETF LWIG working group) выделяет два класса маломощных устройств [88]:

- а) Класс 1: ~ 10 кбайт данных и ~100 кбайт кода,
- б) Класс 2: ~ 50 кбайт данных и ~250 кбайт кода.

(Вычислительная мощность не указана, так как она не является первостепенной характеристикой для текущих реализаций протокола)

В зависимости от приложения, процессоры обычно различаются, начиная от 8-битных микроконтроллеров до маломощных ARM ядер, такие как Cortex-M0. Однако эко-система устройств работающих по протоколу СоAP не ограничена только электронными устройствами с ограниченными ресурсами, но так же включает более производительные устройства, которые играют различные роли.

Так же в рамках данного опроса были идентифицированы следующие роли (от больших до маленьких):

- a) часто приложения, работающие по протоколу СоAP, являются серверными приложениями чаще всего написанными на языках Java или C++ и работающие на мощных машинах.
- б) мобильные устройства, такие как смартфоны и планшеты в основном используются как посредники пользователей для управления электронными устройствами в сети.
- в) некоторые производители специализируются на разработке прокси-устройств, которые играют роль «мостов» между мирами СоAP и HTTP. Этим устройства являются встроенными, но немаленькими вычислительными ресурсами, такие какроутеры и точки доступа.
- г) электронные устройства, обладающие достаточными вычислительными ресурсами, которые являются частями промышленного оборудования, например, автоматизированного оборудования на заводах.
- д) электронные устройства 1-го и 2-го классов используются как сенсоры или простые исполнительные механизмы. Они как правило программируются с помощью языка программирования С.

## **2.3 Методы и алгоритмы нормализации и анализа больших массивов гетерогенных данных генерируемых распределенной сетью электронных потребительских устройств**

### **2.3.1 Методы нормализации и анализа сенсорных данных**

Сети электронных потребительских устройств активно используются в различных областях человеческой деятельности. Их можно найти в большинстве современных мониторинговых систем, включая управление транспортом, мониторинг состояния здоровья, системы обеспечения безопасности, военные системы, мониторинг за окружающей средой и системы, в которых используются механизмы определения местоположения объектов. В таких системах электронные устройства собирают показания о физических феноменах, следовательно, они становятся одним из основных источников потоковых данных.

Такой растущий тренд в использовании распределенных сетей электронных потребительских устройств повышает для программного обеспечения сложность в управлении и сборе показаний с таких устройств [58]. Эту сложность становится более заметной, когда необходимо найти какую-либо информацию в распределенной и гетерогенной сети электронных потребительских устройств, для того чтобы предоставить её в больших объемах, большому количеству пользователей [59]. Такие сети, как правило, состоят из устройств от различных производителей и со специфическими характеристиками. Каждое их электронных устройств предоставляет различные данные, в соответствии с различными схемами данных, с различной точностью и частотой, и в различных единицах измерения. Эта гетерогенность устройств осложняет задачу сбора, как самих данных, так и их метаданных.

Несколько исследовательских работ на данный момент сфокусировано на решение проблемы сбора и анализа данных с распределенной сети электронных потребительских устройств [91, 92, 93, 94]. Эти работы, в общем, направлены на индексирование данных сенсоров, кэширование результатов запросов, и

максимизации количества данных передаваемых по сети одновременно. Хотя они в большей части улучшают производительность алгоритмов поиска и выполнения запросов, но они не в достаточно мере учитывают важность и сложность интеграции гетерогенных сенсорных данных. В отличие от этих работ, работы по управлению семантическими сенсорными данными [64, 65, 66, 67, 68] внедрили различные подходы поиска и вывода на знаниях на основе семантически аннотированных сенсорных данных, в тоже время, учитывая их гетерогенность. Однако, эти подходы все еще не достаточны для того чтобы управлять сенсорными данными в распределенной сети электронных потребительских устройств, и эффективно выполнять запросы в распределенной сети.

### **2.3.2 Моделирование сенсорных данных с помощью онтологий**

Онтологии представляют формальную, применимую и расширяемую модель, которая подходит для представления информации, в нашем случае сенсорных данных, на разных уровнях абстракции и с богатыми семантическими описаниями, которые могут быть использованы для поиска и вывода на знаниях [58]. Более того в условиях гетерогенных данных и устройств, использование стандартов и широко признанных словарей облегчаются задачи публикации, поиска и обмена данными.

Онтологии успешно использовались и используются для моделирования знаний большого количества предметных областей, включая сенсоры и их показаний [69]. Несколько онтологий для представления сенсорных данных были разработаны в последние годы, некоторые из них сфокусированы на описании самих сенсоров, и другие на их показаниях [69]. Большинство из них однако часто сильно связаны с каким-либо определенным проектом или были заброшены, и они как правило не учитывают многие аспекты данной предметной области. Более того многие из них не следуют определенному процессу моделирования или не используют существующие стандарты. Для того чтобы исправить перечисленные недостатки существующих онтологий была сформирована рабочая группа W3C Semantic Sensor Network Incubator Group (W3C SSN), которая предложила общую и независимую от предметной области модель, онтологию SSN

(<http://purl.oclc.org/NET/ssnx/ssn#>), которая в свою очередь совместима со стандартами OGC в части сенсоров и их показаний. Онтология SSN (см. рисунок 2.1) может рассматриваться и использоваться для представления различных свойств элементов реального мира. Например, она может быть использована на описания сенсоров, как они функционируют и обрабатывают внешние возбуждения или сигналы. Или же она может быть главным образом использоваться для представления показаний или измерений и соответствующих метаданных [70]. Например, в статье [71] авторы используют онтологию SSN и соответствующий подход для моделирования крупно сети сенсоров в проекте Swiss Experiment. Например, рассмотрим ветровой сенсор, который расположен в мобильной метеорологической станции. Этот сенсор измеряет скорость ветра в некотором определенном месте. Предположим, что другой сенсор, который расположен в этой же станции и измеряет температуру воздуха каждые 10 минут. В терминах онтологии SSN, измерения (или показания) обоих сенсоров могут быть рассмотрены как показания, каждое из которых со своим природным явлением (ветер и воздух), и каждое из них ссылается на определенное свойство этого явления (скорость и температура).

В онтологии SSN, экземпляры класса Observation представляют такие показания, и они связаны с определенным природным явлением через свойство featureOfInterest, в соответствии с рисунком 2.2.

Таким же образом observedProperty указывает на экземпляр свойства природного явления, такого как скорость. Так как онтология SSN является верхнеуровневой онтологией, то она не пытается определить все возможные типы наблюдаемых свойств. Такие типы свойств могут быть взяты из предметных онтологий, таких как NASA SWEET (<http://sweet.jpl.nasa.gov>).

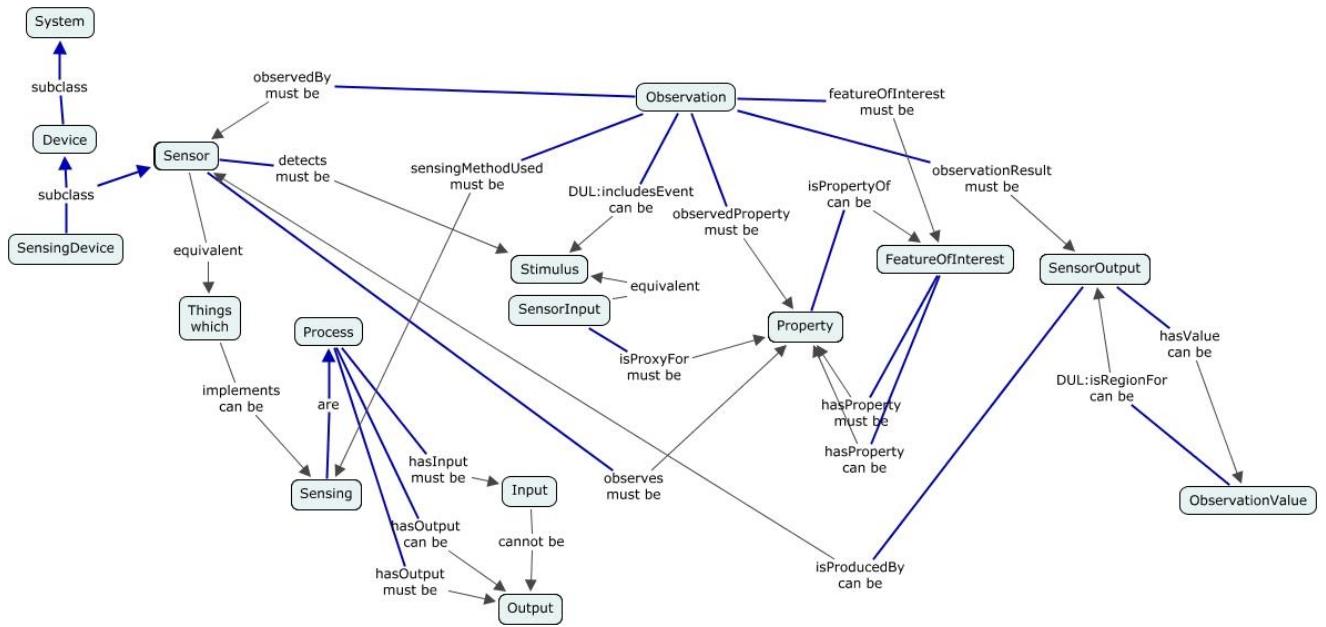


Рисунок 2.1 – Основные элементы онтологии SSN

```

swissex:WindSpeedObservation1 rdf:type ssn:Observation ;
    ssn:featureOfInterest [ rdf:type sweet:Wind ] ;
    ssn:observedProperty [ rdf:type sweetProp:Speed ] ;
    ssn:observationResult [
        rdf:type ssn:SensorOutput ;
        ssn:hasValue [ qudt:numericValue "6.245" ^^ xsd:double ] .
    ];
    ssn:observedBy swissex:SensorWind1 ;

```

Рисунок 2.2 – Пример RDF данных, описывающих показание сенсора

Фактические значения показаний сенсоров могут быть представлены как экземпляры класса SensorOutput через свойство hasValue. Само значение может быть связано через специальное свойство численной онтологии (например, свойство numericValue онтологии Quantities, Units, Dimensions and Data Types ontologies (QUDT)). Наконец само показание может быть связано с определенным сенсором, например сенсор SensorWind1 через свойство observedBy. Очевидно, больше информации о показаниях может быть представлено, включая единицы измерения,

частота, шум, ошибки и т.д. Заметим, что процесс онтологического моделирования требует повторно использовать и комбинировать онтологию SSN с какими-либо предметными онтологиями.

Кроме того в данной статье, авторы моделируют метаданные сенсоров. Например, описывается метеорологическая станция, на которой расположены сенсоры, её географическое местоположение (долгота и широта) с использованием онтологии Basic Geo WGS84 Vocabulary (SG84). На рисунке 2.3 представлено описание местоположения сенсора SensorWind1. Кроме того так же может быть представлена такая информация как ответственный человек, время и дата развертывания и т.п.

```
swissex:SensorWind1 rdf:type ssn:Sensor ;  
    ssn:onPlatform [  
        :hasGeometry [  
            rdf:type wgs84:Point ;  
            wgs84:lat "46.8037166" ;  
            wgs84:long "9.7780305"  
        ]  
    ].
```

Рисунок 2.3 – Пример описание местоположение сенсора

Хотя модель показаний сенсоров предоставляет семантически обогащенное представление данных, сенсоры в основном производят потоки «сырых» данных с отсутствием структуры и поэтому существует некоторая брешь между показаниями и «сырыми» данными. Например, оба сенсора из примера ниже (wan7 и imis\_wfbe) измеряют скорость ветра, но используют разных схемы данных, каждый из них использует разные модели для своих данных, в соответствии с рисунком 2.4.

```
wan7: { wind_speed_scalar_av FLOAT, timed DATETIME }  
imis_wbfe: { vw FLOAT, timed DATETIME }
```

Рисунок 2.4 – Примеры показаний сенсоров

Для того чтобы выполнить запрос на получения текущих показаний скорости ветра в таких условиях, пользователю необходимо знать имена сенсоров, и имена всех атрибутов, которые соответствуют семантическому концепту скорости ветра. Эта задача подвержена ошибкам и её не реально выполнить, если количество сенсоров огромно.

В данной работе используется другой подход, который заключается в описании отображений. Хотя существующих работах предоставляются сенсорные показания и они опубликованы в виде RDF и в соответствии с принципами Linked Data, они не позволяют выполнять запросы на потоках сенсорных данных с точки зрения онтологического моделирования. Авторы идут дальше и предлагают декларативных подход к описанию отображений, которые декларируют, как создавать представления показаний в соответствии с онтологией SSN из сенсорных схем «сырых» данных. Для этих целей используется язык R2RML, разработанный в рамках рабочей группы W3C RDB2RDF. Например, мы можем указать, что для каждого кортежа сенсора *wan7* должен быть создан экземпляр класса *ObservationValue*, как это показано на схеме на рисунке 2.5.

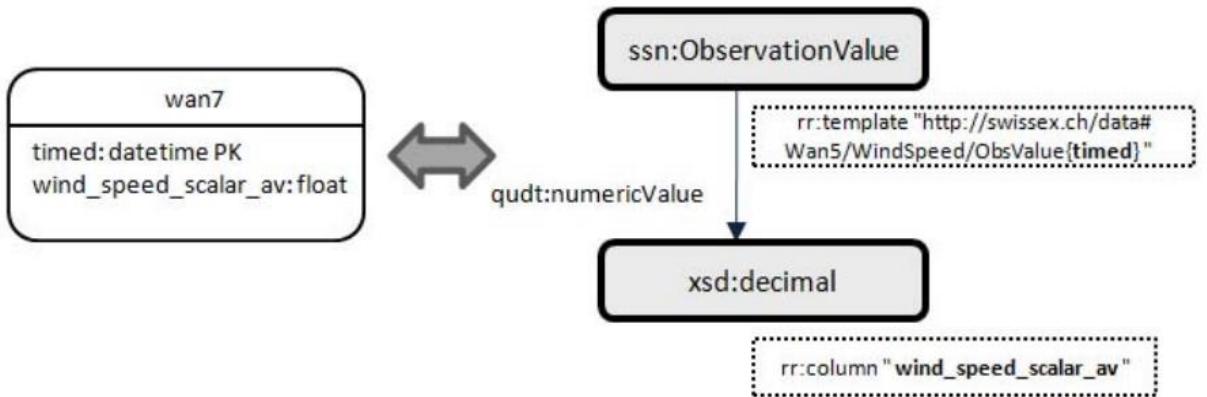


Рисунок 2.5 - Пример простого отображения

И тот же пример представленный на языке R2RML, см. Рисунок 2.6.

Используя такие отображения и онтологию SSN, мы можем описать метаданные сенсора и его показания, с использованием семантической модели, даже если данные лежащие в основе являются реляционными.

```

:Wan7WindMap a rr:TriplesMapClass ;
    rr:tableName "wan7" ;
    rr:subjectMap [
        rr:template "http://swissex.ch/data#Wan5/WindSpeed/ObsValue{timed}" ;
        rr:column "timed" ;
        rr:class ssn:ObservationValue ;
        rr:graph swissex:WannengratWindSpeed .
    ] ;
    rr:predicateObjectMap [
        rr:predicateMap [
            rr:predicate qudt:numericValue
        ];
        rr:objectMap [ rr:column "wind_speed_scalar_av"
    ] .

```

Рисунок 2.6 – Пример правила отображения показаний сенсора

## **2.4 Методы и алгоритмы интерактивной визуализации больших массивов гетерогенных данных**

Сегодня, благодаря широкому распространению технологий сети Интернет, становится все проще работать с большими объемами данных. Множество открытых наборов данных публикуются порталами открытых данных на самые различные темы – транспорт, качество воды и воздуха, социологические и культурологические исследования – обычно в формате CSV или таблиц; иногда публикуются дампы обычных реляционных баз данных. Однако, такие данные как правило не являются адаптированными (согласно принципам Linked Open Data – LOD [102]) для машин. Тим Бернерс-Ли предложил так называемую пятизвездную модель, в которой данные возможно оценить по шкале интероперабельности от 1 (неструктурированные, «сырые» данные) до 5 (семантически связанные данные). Для повышения качества данных по пятизвездной системе был разработан RDF (Resource Description Framework) – унифицированный язык для описания высказываний о данных в формате триплетов в соответствии с принципами LOD, позволяющий связывать между собой различные типы данных и получать к ним доступ через сеть Интернет.

Одним из главных препятствий для массовой публикации открытых данных за пределами сообщества Semantic Web является техническое и концептуальное обоснование преимуществ составления и предоставления доступа к наборам данных. Интуитивная, емкая и выразительная визуализация таких данных решает эту проблему, побуждая пользователя взаимодействовать и исследовать данные, представленные в наиболее удобном для восприятия формате.

Бен Шнейдерман [73] выделил семь основных типов, по которым может быть классифицирован атомарный фрагмент данных:

- a) Одномерные. Линейные типы данных, такие как текстовые документы, исходные коды программ, различные сортированные списки. Такие данные обычно отображаются в виде определенной последовательности

согласно одному критерию (например, в алфавитном порядке) и, как следствие, обычно не подлежат визуализации в виду своей простоты. Исключение – повторяющиеся через весь датасет структуры, такие как названия месяцев, которые могут быть агрегированы для дальнейших шагов распределенного анализа.

- б) Двумерные. Картографические данные, планы, чертежи. Благодаря широкому применению движков для создания интерактивных карт, визуализация двумерных данных позволяет наглядно продемонстрировать паттерны и соответствия, имеющиеся в данных; применяются такие техники, как кластеризация объектов и тепловые карты (heatmaps). При интеграции с одномерными данными позволяют получить более емкие и наглядные визуализации с описаниями, изображениями и т. д.
- в) Трехмерные (объемные). Обычно представляют объекты реального мира молекулы, тела, строения, а также сложные взаимоотношения между ними. Одни из самых востребованных в области научных визуализаций. Такие технологии, как WebGL, делают качественные и сложные трехмерные визуализации доступными для рядовых пользователей.
- г) Многомерные. Как правило – данные, полученные из реляционных и статистических баз данных с N атрибутами. В зависимости от задачи и размерности визуальной презентацией являются различные точечные графики и диаграммы.
- д) Временные. В отличие от одномерных данных, данные имеют временное начало и конец. Полезны при необходимости отображения хронологии или регулярности на календарях или так называемых временных линиях (от англ. timeline).
- е) Иерархические. Содержат элементы, связанные с родительскими (кроме корневого) и образующие древовидные структуры. Лучше всего подходит для наглядного представления структур в целом, таких, например, как файловое дерево операционной системы, фамильное

дерево или граф связанных классов в какой-либо онтологии. Для визуального представления обычно используются дендограммы, диаграммы Вороного, а также Icicle-диаграммы.

ж) Сеть. Представляет связанные данные, не имеющие четко выраженной иерархии.

Сформулированный Шнейдерманом и популяризованный им в качестве “мантры информационной визуализации” принцип “Сначала общий обзор, затем приближение, фильтрация и детали по необходимости” отражает базовые концепты, применяемые при пользовательском взаимодействии с наборами данных. В таблице 2.2 приведено сравнение популярных типов графиков на основе типа применяемого анализа.

Таблица 2.2 - Анализ популярных графиков по типу анализа

	Сравнение	Композиция	Распределение	Отношение
Линейный график	*			
Гистограмма	*		*	
Bullet-гистограмма	*			
Составная гистограмма		*		
Областной график		*		
Составной областной график		*		
Круговая диаграмма		*		
Диаграмма разброса			*	*
Пузырьковая диаграмма	*		*	*
Каскадная диаграмма		*		

В 2011 году, Аба-Сах Дадзи и Метью Роу провели исследование современных подходов к визуализации связанных данных, в котором были представлены следующие академические или прикладные инструменты:

a) CODE Visualisation Wizard. Разработан в рамках основанного в ЕС исследовательского проекта CODE (<http://code-research.eu>), позиционируется как облачная платформа для визуализации. Основывается на словаре RDF Data Cube Vocabulary [104], разработанном W3C для представления статистических данных. Платформа предоставляет встроенный парсер данных в DCV-формат [73], интерактивный выбор размерности и типов визуализации, а также сохранение истории действий, что делает процесс анализа воспроизводимым. Результат работы CODE VW представлен на рисунке 2.7.

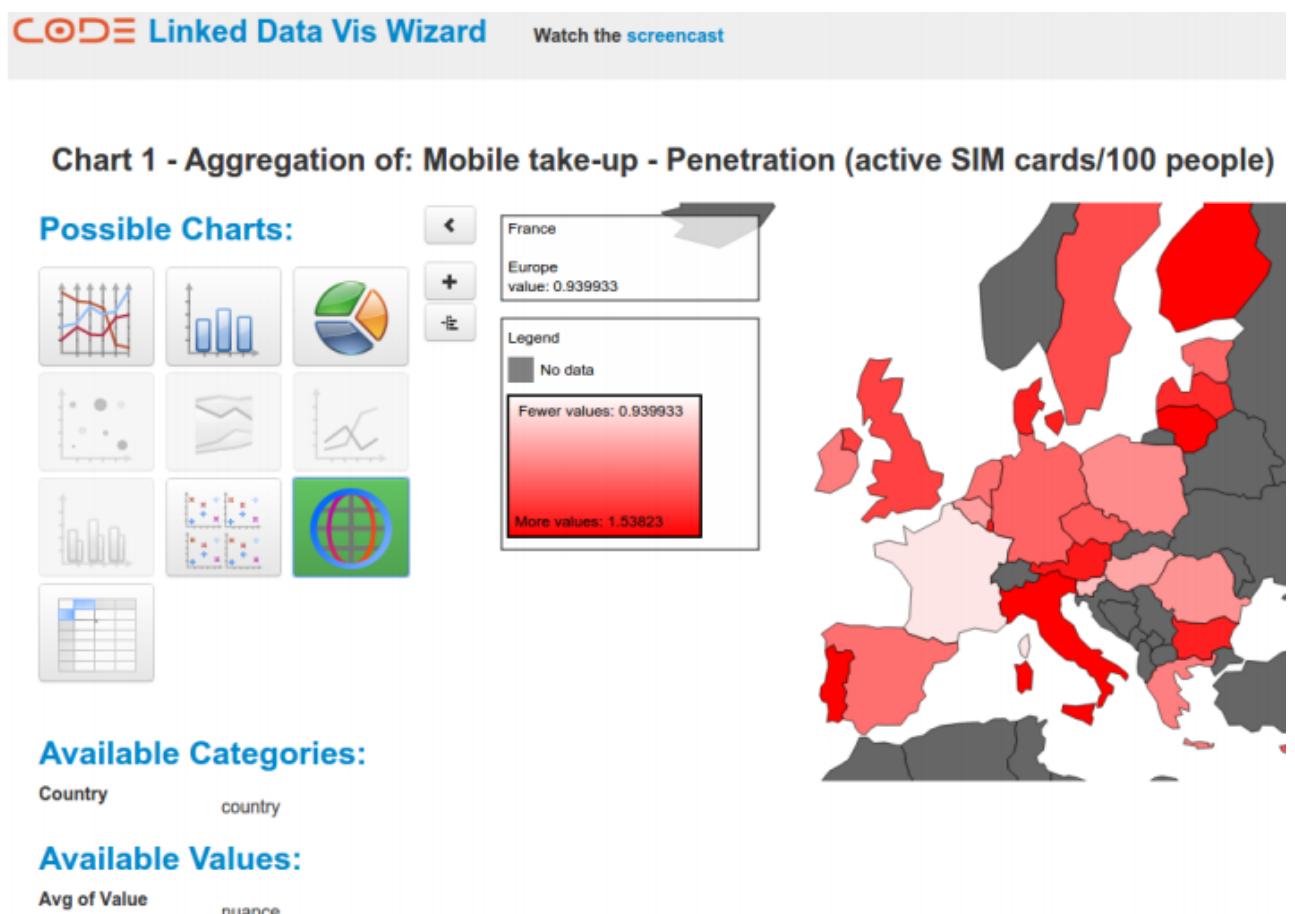


Рисунок 2.7 - Платформа CODE Visualisation Wizard

б) LDVizWiz [75] позволяет визуализировать любые стандартизованные SPARQL endpoint'ы, а также проводит первичную интроспекцию данных на основе классификации Шнейдермана, самостоятельно выполняя запросы и распределяя данные по следующим категориям:

географические, временные, события, люди, организации, статистика и знания. Пример работы приложения представлен на рисунке 2.8.

The screenshot shows the LDVizWiz application interface. At the top, there is a navigation bar with links to 'Linked Data Visualization Wizard', 'Home', 'About', and 'Contact'. Below the navigation bar, there is a horizontal menu with tabs: 'GeoData' (which is selected and highlighted in blue), 'ORG Data', 'Stat Data', 'Person Data', and 'Event Data'. The main content area has a title 'Generic view (tabular)' followed by a sub-title '(100 triples)'. Below this, there is a table with four columns: 'lat', 'long', 'name', and 'resource'. The table lists five events with their coordinates and names. The last row of the table is highlighted in light blue. Below the table is a map view showing the locations of these events. A callout box on the map provides details for the event 'Innovation Forum: The Internet: The Next 40 years' located at approximately 51.5128 lat and -0.13342 long. The callout box contains the event name and its resource URL (<http://data.linkedevents.org/event/064b358b-542e-435e-94a1-4e8966da162f>). The map also shows other locations like Brent Park, Westbere Copse, Barnsbury Wood, Stratford, Marylebone, and The British Museum.

lat	long	name	resource
37.7794	-122.398	[IxDA SF Presents] High Fidelity Holiday: an interactive seasonal celebratory experience aka PARTY	<a href="http://data.linkedevents.org/event/0615d234-29b8-4fce-9bbf-d9ef9db4b001">http://data.linkedevents.org/event/0615d234-29b8-4fce-9bbf-d9ef9db4b001</a>
40.3979	-111.932	Utah WordPress Meetup	<a href="http://data.linkedevents.org/event/063df25d-539b-445e-92ed-ff2ebde55736">http://data.linkedevents.org/event/063df25d-539b-445e-92ed-ff2ebde55736</a>
51.5128	-0.13342	Innovation Forum: The Internet: The Next 40 years	<a href="http://data.linkedevents.org/event/064b358b-542e-435e-94a1-4e8966da162f">http://data.linkedevents.org/event/064b358b-542e-435e-94a1-4e8966da162f</a>
53.2695	-113.141	Elizabeth May tours Alberta July 8th Events	<a href="http://data.linkedevents.org/event/0655a494-a6ca-4e15-84ec-894204070550">http://data.linkedevents.org/event/0655a494-a6ca-4e15-84ec-894204070550</a>
29.721	-95.4184	Houston Singles Zydeco Party!	<a href="http://data.linkedevents.org/event/069a11e3-b077-4254-8d40-f9580b555aa1">http://data.linkedevents.org/event/069a11e3-b077-4254-8d40-f9580b555aa1</a>

Рисунок 2.8 - Платформа LDVizWiz

- в) LODVizSuite & ResXplorer. Ученые из центра цифровых исследований iMinds разработали, опираясь на принципы EDA (от англ. Exploratory Data Analysis [76]) и Шнейдермана, собственный интерактивный визуальный рабочий процесс, а также инструмент для совместной работы с данными по этим принципам, который был назван LODVizSuite (Рисунок 2.9).

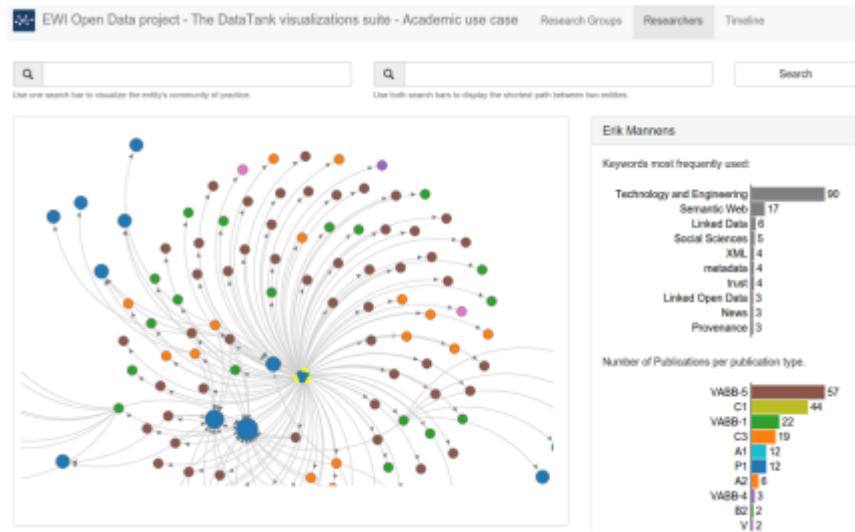


Рисунок 2.9 - Инструмент LODVizSuite

г) LODVisualization разработан на основе, так называемой модели визуализации связанных данных (LDVM - Linked Data Visualization Model [77], Рисунок 2.10). Инструмент предназначен для работы с общим Обзором (по Шнейдерману), представляя иерархию классов и свойств, определенных в SPARQL - графе. Позволяет отрисовывать деревья, таблицы и гистограммы, основываясь на любом открытом SPARQL endpoint.

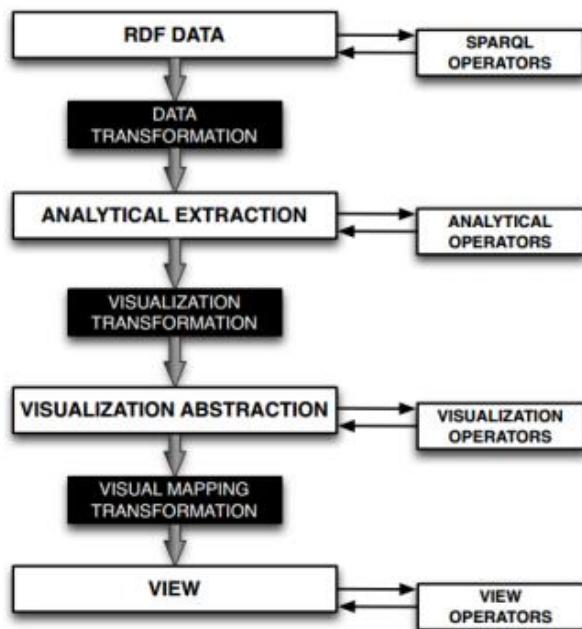


Рисунок 2.10 - LDVM – модель

e) Payola [78] представляет собой изящную реализацию модели LDMV для генерации визуальных представлений чешского облака LOD - набора публичных наборов данных Чехословакии. Также предоставляет интерфейсы для работы со SPARQL или RDF-файлами, а также пользовательский веб-интерфейс. Предоставляет расширяющую систему плагинов. На рисунке 2.11 представлен результат запроса к одному из открытых чешских датасетов.



Рисунок 2.11 - Пользовательский интерфейс Payola

д) rdf:SynopsViz. Основной упор в данном инструменте был сделан на метаданные датасетов, позволяющие выявить неявную внутреннюю структуру. Предоставляет обширные инструменты для подсчета общей статистики - разброса, минимумов и максимумов, значимости и т. д. Скриншот работы программы представлен на рисунке 2.12.

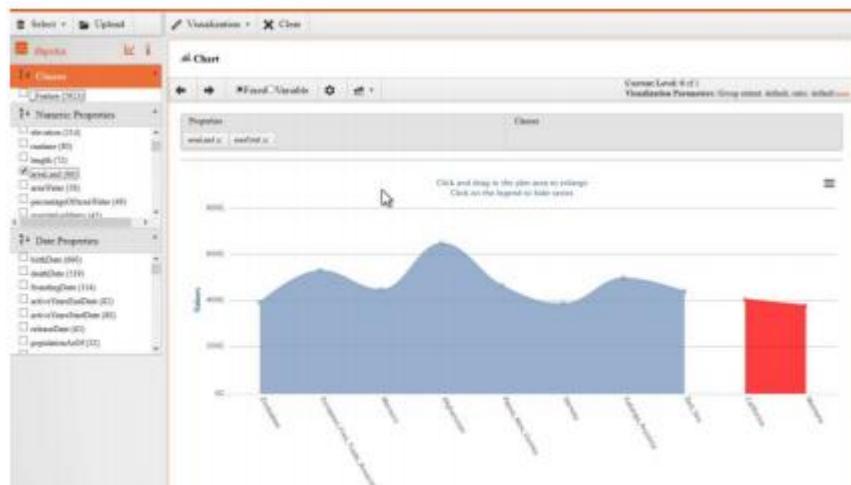


Рисунок 2.12 - Пользовательский интерфейс rdf:SynopsViz

- e) Sgvizler представляет собой обертку на языке JavaScript для визуализации результатов SPARQL - запросов в виде встроенных элементов на любых веб-сайтах. Используя принципы HTML5, предоставляет на выходе семантическую разметку на основе data-аттрибутов в соответствии с выполненным запросом. Автоматически определяет тип визуализации и размерность на основе запроса. На рисунке 2.13 представлен текстовый и графический форматы результатов запроса к системе. Sgvizler имеетстроенную интеграцию с такими популярными JavaScript-библиотеками, как Google Charts и Data-Drive Documents (D3), а также позволяет делать CORS-запросы [109] на другие домены. Тем не менее, этот инструмент требует от пользователя некоторого знания технологий Semantic Web, для того, чтобы правильно сформулировать нужный SELECT-запрос.

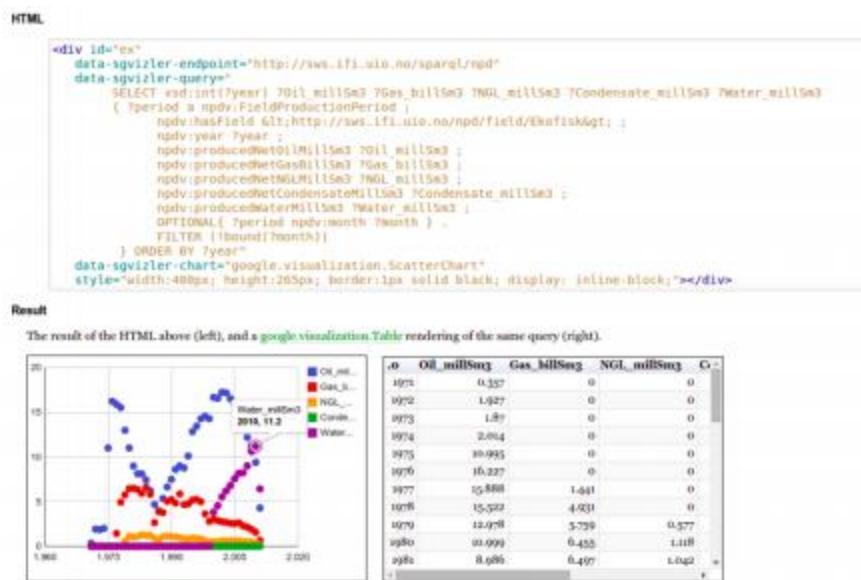


Рисунок 2.13 - Результат работы Sgvizler в текстовом и браузерном представлениях

ж) Visualbox, как и Sgvizler, является инструментов для людей, имеющих некоторую техническую квалификацию. Представляет собой полноценный редактор SPARQL-запросов с подсветкой синтаксиса и ошибок, а также возможностью выполнения этих запросов и визуализации результатов, в соответствии с рисунком 2.14. Распространение результатов также поощряется; у каждой визуализации имеется свой уникальный URI.

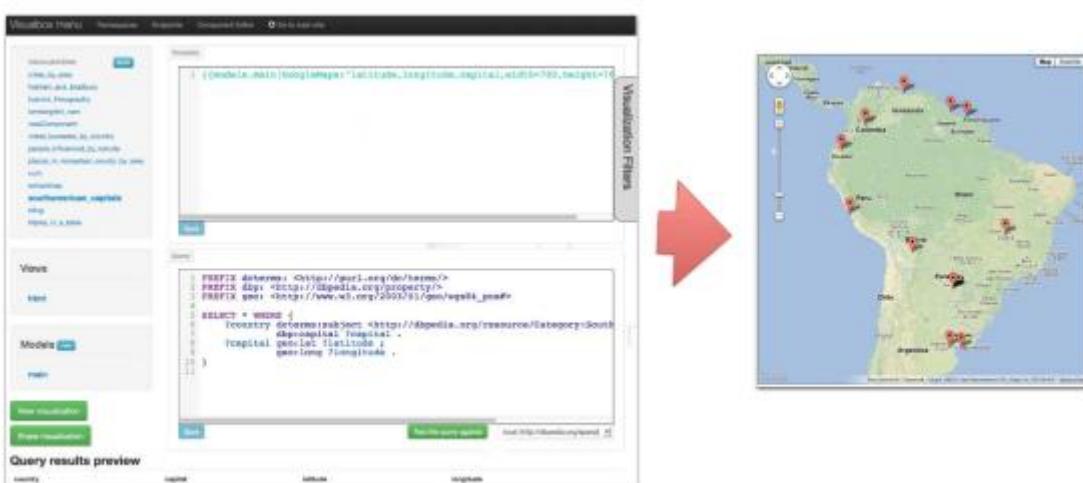


Рисунок 2.14 - Визуальный редактор Visualbox и графическое представление результатов

и) VizBoard позиционируется как рабочая среда на основе платформы CRUIS (Composition of Rich User Interface Services [80]). Пользователи имеют возможность смешивать различные размерности данных для создания более комплексных и полных представлений наборов данных, в соответствии с рисунком 2.15. Поддерживает всю информацию о рендеринге визуализаций, используя специализированный словарь VISO, описывающий различные отношения между свойствами.

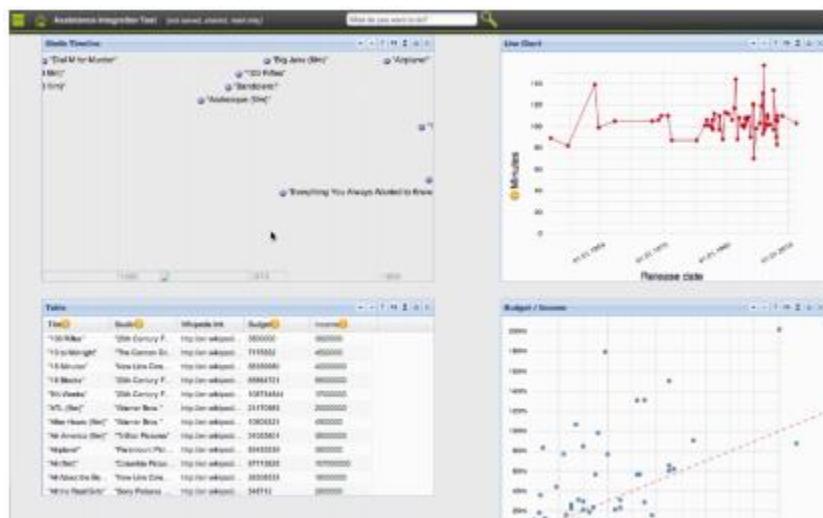


Рисунок 2.15 - Пользовательский интерфейс VizBoard

В таблице 2.3 и таблице 2.4 представлен сравнительный анализ рассмотренных инструментов по различным критериям.

Таблица 2.3 - Сравнение инструментов визуализации по типу данных

Название ПО	1-D	2-D	3-D	N-D	Временные	Иерархические	Сеть
CODE		*		*			*
LDVizWiz		*		*	*		*
LOD/VizSuite					*		*
LODVisualization		*				*	

### Продолжение таблицы 2.3

Название ПО	1-D	2-D	3-D	N-D	Временные	Иерархические	Сеть
Payola		*		*		*	*
rdf:SynopsViz				*	*	*	
Sgvizler	*	*		*	*	*	*
VisualBox		*		*	*	*	*
VizBoard				*	*		*

Таблица 2.4 - Сравнение инструментов визуализации по решаемым задачам

	Обзор	Приближение	Фильтр	Детали	Отношения	История
CODE	*	*	*		*	*
LDVizWiz	*	*		*		
LOD/VizSuite	*	*			*	
LODVisualization	*					
Payola	*	*		*		
rdf:SynopsViz	*	*	*			
Sgvizler	*	*	*	*	*	
VisualBox	*	*	*	*	*	
VizBoard	*		*	*	*	

## **2.5 Методы передачи данных от потребительских устройств к центрам обработки данных**

### **2.5.1 Обзор протоколов передачи данных**

Существует множество работ в области построения беспроводных сенсорных сетей для обеспечения передачи данных между электронными потребительскими устройствами и центрами обработки данных. Были разработаны протоколы обеспечивающие передачи данных в беспроводной сети электронных устройств с ограниченными ресурсами. Существуют две стороны проблемы. Первая – сенсоры сами по себе имеют ограниченные ресурсы, и они подключены к сети Интернет напрямую или же через коммуникационный шлюз. Поэтому подключение к сети Интернет или коммуникационному шлюзу является проблемой. В данном случае коммуникационный шлюз не ограничен в ресурсах. С другой стороны, обнаруживается новая проблема, если сам коммуникационный шлюз имеет ограниченные ресурсы, например, смартфон или планшет, через который обмениваются информацией большое количество сенсоров. Для того чтобы обеспечить низкое потребление энергии для коммуникации, большинство сенсоров используют модули построенные по спецификации ZigBee. Так же использование протокола IPv6 в таких сетях с низким потреблением энергии, привело к созданию стандарта 6LoWPAN [111]. Протокол 6LoWPAN обеспечивает беспроводное соединение с помощью протоколов IEEE 802.15.4 и IPv6 простым и понятным способом.

Для обеспечения работы программного обеспечения с сенсорами с ограниченными ресурсами на прикладном уровне, были предложены для протокола передачи данных: MQTT от компании IBM и CoAP от организации IETF. Протокол CoAP основан на модели запроса-ответа. В статье [82] демонстрируется, как CoAP поверх протокола UDP может быть лучшим выбором, чем HTTP поверх TCP или HTTP поверх UDP, если принимать во внимание требования к низкому энергопотреблению, избыточности заголовков и стабильность на прикладном

уровне. Протокол MQTT использует метод публикации-подписки [113] и иерархию тем подписок, а также позволяет использовать его сенсорам с ограниченными ресурсами через метод инициализации передачи данных «PUSHing» [114] от сервера, вместо того чтобы сенсорам опрашивать сервер с некоторым интервалом для получения данных с сервера. Однако, в контексте коммуникации с конечными сенсорами беспроводной сети, IBM разработали другой протокол передачи данных, называемый MQTT for Sensors (MQTT-S) [115], который не привязан к сетевым сервисам.

Как можно заметить во множестве автомобилей, медицинском оборудовании (например, [116], [117]), в большинстве случаях используются проприетарный и нестандартный механизм коммуникации, часть которого скрыта от разработчика. Не скрытой же частью остается механизм коммуникации от коммуникационного шлюза к центрам обработки данных. Поэтому специфичный для сенсорных сетей протокол сетевого уровня, такой как 6LoWPAN тут не рассматривается. Также, прикладные протоколы, такие как MQTT-S не подходят, так как разработаны главным образом для коммуникации между сенсорами в одной беспроводной сети.

### **2.5.2 Поддерживаемые возможности протоколов передачи данных**

В данном подразделе перечисляются возможности/функциональности, поддерживаемые протоколами передачи данных CoAP и MQTT.

CoAP является протоколом передачи данных в сети Интернет, основанный на архитектуре REST [118]. Он очень похож на протокол HTTP. Однако чтобы избежать излишнюю нагрузку, которая существует в протоколах, базирующихся на TCP, CoAP изначально спроектирован для использования поверх UDP, который реализует ограниченный механизм гарантированной доставки сообщений из протокола TCP. Необязательный механизм гарантированной доставки сообщений реализован на уровне сообщений протокола UDP. Логически, CoAP может рассматриваться как набор из двух логических уровней: уровень запрос/ответ и уровень сообщений для взаимодействия с UDP. Он поддерживает четыре категории

сообщений: подтверждение, не подтверждение, подтверждение о получении и сообщение сброса. Сообщения категорий подтверждения необходимы для реализации механизма гарантированной доставки сообщений.

В отличие от HTTP, СоAP поддерживает асинхронный механизм доставки сообщений. СоAP имеет меньшую избыточность и сложность разбора сообщений, чем HTTP. Хотя изначально он разработан для использования поверх UDP, но он также может использоваться поверх TCP [119].

Основную версию протокола СоAP расширяли для реализации поддержки режима публикации/подписки в целях поддержки такой функциональности как получение обновлений в режиме реального времени, как было предложено в статье [83].

MQTT, как было упомянуто выше, использует архитектуру публикации/подписки с иерархией тем подписки и с агентом обмена сообщениями, который объединяет публикующих и подписчиков. Он поддерживает три уровня качества сервиса для доставки сообщений с повышенной гарантией доставки и имеет низкую избыточность. MQTT ориентирован на поддержку постоянного соединения с агентом обмена сообщениями для обеспечения коммуникации публикующих и подписчиков. Он основывается на TCP.

На рисунке 2.16 представлено графическое отображение протоколов на OSI модель.

В таблице 2.5 представлено сравнение возможностей этих протоколов и потенциальные варианты их использования.

Так как СоAP использует запросы и ответы, похожие на HTTP и повторяет его в нескольких местах, то СоAP может использоваться для отправки GET, PUT, POST, DELETE запросов. Поэтому он может быть использован как протокол для навигации. Потенциальные приложения могут использовать специальные прокси для конвертации СоAP запросов в HTTP.

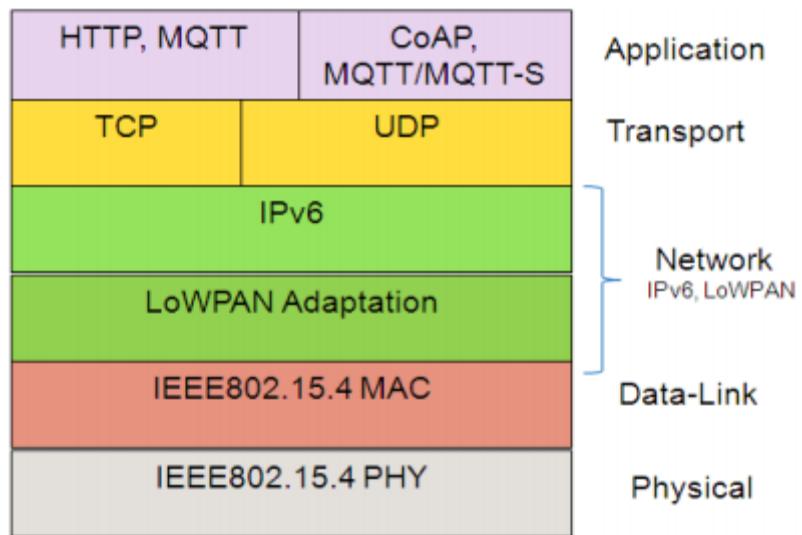


Рисунок 2.16 - Отображение протоколов на OSI модель

Таблица 2.5 - Сравнение протоколов MQTT и СоAP

Протокол	Транспортный уровень	Режимы	Структура заголовков	Таймаут ответа	Макс. количество повторов	Поддержка multicast	Надежность
MQTT	TCP	публикация/подписка	2 байта – заголовок	настраиваемый	настраиваемый	да	3 уровня качества сервиса
СоAP	UDP, но может быть и TCP	REST запросы/ответы, публикация/подписка	4 байта - заголовок	2 секунды, настраиваемый	4 и 0 для multicast	да	повтор с таймаутом

MQTT, в силу своей архитектуры публикации/подписки, очень полезен для реализации приложений использующих push модель вместо poll модели. Однако в случае коммуникационного шлюза с ограниченными ресурсами этот протокол может быть использован для публикации сенсорные данные в режиме реального времени на веб-сервер асинхронно и надежно. Попытки интегрировать MQTT с системой Pachube, которая является широко известной технологией мониторинга

сенсоров и устройств, используя механизм опросов, Pachube может использовать возможности MQTT для настройки триггеров.

Однако, СоАР получил более широкое применение для устройств с ограниченными ресурсами, так как упрощает интеграцию с 6LoWPAN, легко конвертируется в HTTP, поддерживает UDP операции с низким уровнем избыточности, поддержка засыпания узлов сети, поддержка одновременно режима запроса/ответа и публикации/подписки и т.д.

## **2.6 Технические принципы и подходы к организации защиты информации передаваемой в распределенной сети электронных потребительских устройств (Internet of Things)**

### **2.6.1 Обзор методов обеспечения безопасности в распределенной сети электронных потребительских устройств**

Хотя исследования безопасности технологий Интернета вещей находятся на ранней стадии, существует несколько серьезных работ по анализу существующих проблем и потенциальных механизмов защиты. Однако, существующие работы больше сфокусированы на обзоре существующих проблем обеспечения безопасности, не принимая во внимание, то как каждая из возможностей Интернета вещей влияет на уровень безопасности. Для того чтобы понять определенные проблемы безопасности распределенной сети электронных потребительских устройств, необходимо не только проанализировать вклад принципов Интернета вещей в существующие модели угроз и атак, но также изучить влияние этих принципов на интеграцию различных механизмов обеспечения безопасности.

Одной из главных исследовательских проблем, которые влияют на развитие технологий Интернета вещей, является обеспечение необходимого уровня безопасности. Архитектура технологий Интернета вещей должна справляться с миллиардами устройств, которые будут взаимодействовать друг с другом. И все эти взаимодействия должны быть защищены некоторым образом, обеспечивая

безопасность передачи информации и предоставления сервисов, для всех участников, тем самым снижая количество инцидентов.

Однако обеспечение безопасности Интернета вещей является сложной задачей. Существует большое количество угроз, которые могут повлиять на участников процесса, такие как угрозы на каналы передачи данных, физические атаки, атаки на отказ, подмена идентификации устройств и другие [90]. И наконец, неотъемлемая сложность архитектуры Интернета вещей, где множество гетерогенных устройств расположенных в разных контекстах могут обмениваться информацией друг с другом, что еще больше осложняет проектирование и развертывание эффективных, масштабируемых механизмов обеспечения безопасности.

Некоторые из перечисленных выше исследовательских проблем были уже определены научным сообществом [121, 122, 48]:

- a) Гетерогенность имеет сильное влияние на сервисы обеспечения безопасности на уровне протоколов и сетей, которые должны быть реализованы в Интернете вещей. Устройства с ограниченными ресурсами будут взаимодействовать с разными гетерогенными устройствами (например, другие устройства с ограниченными ресурсами, развитые веб-сервисы) либо напрямую, либо шлюзы. В данном случае, важно не только реализовать эффективные криптографические алгоритмы, которые могут предоставлять высокую пропускную способность даже на 8-битных или 16-ти битных устройствах, но так же адаптировать или создать легковесные протоколы безопасности, которые предоставляют законченный коммуникационный канал. Эти протоколы требуют сертификатов доступа, поэтому должны быть разработаны оптимальные системы управления ключами для предоставления доступа к сертификатам доступа.

- б) Существование миллиардов гетерогенных устройств так же оказывается на управлении идентификацией. Помимо определения фактических границ «идентификации» в данном контексте, нам также необходимо предоставить некоторый механизм для достижения универсальной аутентификации. Без аутентификации, будет невозможно убедиться в том что некоторый поток данных произведен некоторым устройством и содержит, то что должен содержать. Другим важным аспектом связанным с аутентификацией является авторизация. Если нет контроля доступа к чему либо, то все будет доступно всем, что практически не осуществимо и не реалистично.
- в) В самом деле, огромный поток данных, вызванный миллиардами сущностей создающими информацию - это большая угроза для конфиденциальности. Пользователи должны иметь инструменты, которые позволяют им сохранить свою анонимность в таком «супер связанном» мире. Другие инструменты должны предоставить «снимок» информации и политики, связанных с конкретным пользователем, которые обеспечат прозрачность и предотвратят убеждение, что Интернет вещей тихо контролирует нашу жизнь. В самом деле, технологии Интернета вещей сама должна серьезно рассмотреть вопрос о реализации в частную жизнь, принципы проектирования [123], обеспечивая, ориентированного на пользователей, поддержка безопасности и конфиденциальности от его собственных фондов.
- г) Размеры и гетерогенность Интернета вещей также влияют на доверие и власть. Можно выделить два типа доверия: (а) доверие во взаимодействии между элементами, где мы должны справляться с неопределенностью будущих действий всех взаимодействующих элементов. И (б) доверие к системе со стороны пользователя, где пользователь должен иметь возможность управлять собственной информацией, чтобы не быть под внешним контролем.

д) Количество уязвимых систем и методов атак определено будет увеличиваться в контексте Интернета вещей, поэтому отказоустойчивость является важным элементом. Мы должны бороться за обеспечение должного уровня безопасности по умолчанию, и также мы должны разрабатывать механизмы информирования, которые могут быть использованы для создания основы для обнаружений внедрений и механизмов их предотвращения, которые помогут элементам Интернета вещей защищаться или даже отключаться в случае атаки. И наконец, элементы Интернета вещей должны уметь перенаправлять функциональность в обход небезопасных зон.

### **2.6.2 Анализ моделей угроз и атак**

Для того чтобы понять как обеспечить безопасность различных подходов рассмотренных выше в будущем, в первую очередь должны быть перечислены и проанализированы модели атак. В статье [95] перечисляются следующие модели атак:

- а) Отказ в обслуживании (DoS). Существует широкий набор атак, которые могут быть применены к распределенной сети электронных потребительских устройств. Кроме традиционных атак в сети Интернет на отказ в обслуживании, когда ресурсы сервис провайдеров или коммуникационный канал исчерпаны, так же может быть атакован беспроводной канал связи сети.
- б) Физическое повреждение. Такой тип атаки может рассматриваться как один из типов атак на отказ в обслуживании. Это вполне вероятная атака в контексте Интернета вещей, так как устройства могут быть легкодоступны любому (например, уличный фонарь).
- в) Перехват информации («прослушка»). Пассивный взломщик может нацелиться на различные коммуникационные каналы (напр., беспроводные сети, локальные проводные сети, Интернет) для того чтобы получить доступ к данным из информационного потока.

Очевидно, что взломщик, который получает доступ к определенной инфраструктуре, может получить доступ к данным, которые существуют в этой инфраструктуре.

- г) Захват информации узла. Как сказано выше электронные потребительские устройства физически размещаются в некоторых условиях. Вместо того, чтобы уничтожить их, взломщик может получить доступ к информации, хранящейся на данном устройстве.
- д) За получение контроля. Активный взломщик может заполучить частичный или полный контроль над элементом Интернета вещей. Степень опасности зависит главным образом от важности данных обрабатываемых данным элементом и от важности задачи, которая выполняется данным элементом.

### **2.6.3 Конкретные проблемы и потенциальные решения**

#### **2.6.3.1 Идентификация и аутентификация**

Важным является то, как организуется идентификация и аутентификация в распределенной сети электронных потребительских устройств, так как множеству устройствами необходимо аутентифицировать друг друга для того чтобы обеспечить работы доверенных сервисов [96]. Определяя такие механизмы обеспечения безопасности, мы так же должны учитывать некоторые неотъемлемые характеристики Интернета вещей. Так как взаимодействие между узлами сети достаточно динамичное, то узлы сети могут даже не знать заранее с кем они должны взаимодействовать для обеспечения некоторой функциональности. Электронные потребительские устройства, установленные на автотранспорте являются таким примером, когда ожидается, что автомобиль должен предоставлять доступ к данным не только устройствам внутри себя, но и другим автомобилям. Кроме того, если миллиарды электронных потребительских устройств будут взаимосвязаны, то необходимо обеспечивать масштабируемость механизмов их идентификации.

Масштабируемость механизмов уникальной идентификации электронных потребительских устройств важна и к 2012 году существуют различные механизмы, которые могут быть использованы для уникальной идентификации устройств, такие как коды тегов стандартов EPC и ucode [97]. Поэтому в будущем ожидается, что устройства будут сосуществовать ни только на универсальном уровне, но и на локальном уровне [98].

Что касается аутентификации устройств, то должны быть рассмотрены несколько сценариев, когда устройства относятся к какой-либо группе расположенной в одном зоне. В таких условиях, локальные провайдеры идентификации могут обеспечивать аутентификацию таких устройств, и также могут создавать круг доверия с релевантными внешними провайдерами. Соответственно, локальные устройства могут быть аутентифицированы не только внутри группы, но и когда они взаимодействуют с внешними устройствами. Такая групповая стратегия частично была для взаимодействия внутри беспроводной сенсорной сети, где взаимодействие за пределами группы возможно благодаря распределенной сети провайдеров идентификации [128]. Традиционные методы аутентификации, применяемые в сети Интернет так же могут быть использованы, например, OpenID и Shibboleth, но надо учитывать, что они не были спроектированы, чтобы отвечать всем требованиям Интернета вещей [100].

### **2.6.3.2 Контроль доступа**

В рамках Интернета вещей исследовательские проблемы, связанные с обеспечением контроля доступа сильно похожи на те, что существуют в любых других распределенных системах. Определенный сервис состоит из комбинации нескольких других сервисов и источников данных из разных местоположений и контекстов. Все эти провайдеры информации имеют свой собственный механизм обеспечения доступа, чей жизненный цикл необходимо поддерживать.

Существуют так же специфические проблемы, которые должны учитываться в контексте Интернета вещей. Гранулярность, т.е. предоставление большей

информации пользователи с валидными правами доступа, и местоположение становятся важными элементами обеспечения контроля доступа в некоторых сценариях.

Существует только несколько работ по обеспечению контроля доступа в распределенной сети электронных потребительских устройств. Так как применения существующих подходов не является простой задачей. Например, существуют проблемы с масштабируемостью и непротиворечивости в хранении списков пользователей и соответствующих им прав доступа. Контроль доступа на основе ролей требует определения различных ролей пользователя в различных контекстах. И наконец, контроль доступа на основе ролей использует сертификаты атрибутов [101], для которых требуется инфраструктура, способная проверять такие сертификаты различных условиях. Однако, благодаря наличию специфических характеристик, есть возможность рассматривать некоторые факторы, такие как контекст, частью механизма контроля доступа [102].

### **2.6.3.3      Безопасность протокола и сети**

Безопасный канал передачи данных, в большинстве случаев, является побочным результатом успешной аутентификации. Этот процесс использует некоторые сертификаты доступа пользователя, такие как X.509 сертификаты. Если существуют некоторый набор известных проверенных провайдеров, то распределение и управление этими сертификатами становится проще, так как появляется возможность подгрузить необходимую информацию. Однако в распределенной сети появляются дополнительные проблемы: любой устройство сети может взаимодействовать с любым другим устройством и эти устройства могут не знать друг друга заранее. Поэтому в таком сценарии управление ключами доступа является проблемой.

Так как Интернет вещей проникает в экосистему сети Интернет, то важно предоставить поддержку существующих протоколов безопасности. На самом деле безопасность протоколов передачи данных, таких как СоАР в большинстве зависит

от реализации таких протоколов безопасности [103]. Некоторые протоколы могут быть реализованы без серьезных модификаций. Например, существуют коммерческие реализации протокола DTLS для устройств с ограниченными ресурсами [133]. Однако другие протоколы необходимо адаптировать из-за сложности их архитектуры.

#### **2.6.3.4 Конфиденциальность**

До сих пор распределенная архитектура сети электронных потребительских устройств требовала реализации только более сложных механизмов обеспечения безопасности. Однако существует одна область, которая преподносит некоторые достоинства: управление данными и конфиденциальность. Базовая идея заключается в том, что каждый узел сети имеет больший контроль над данными, которые он генерирует и обрабатывает. Существует несколько последствий такого подхода. Во-первых, устройства могут контролировать гранулированность данных, которые они производят. Например, портативный сенсор радиации может анонсировать, что он находится в определенной области, но не сообщить свои точные координаты. Во-вторых, узлы сети могут определять свои собственные политики контроля доступа. В-третьих, узлы не должны предоставлять все данные, которые они производят, внешним узлам, а только те данные, которые необходимы для реализации определенного сервиса.

Подход распределенной сети электронных потребительских устройств пользуется реализацией принципов умышленной конфиденциальности, так как все элементы сети могут управлять своими собственными данными. Также необходимо изучить возможность применения существующих алгоритмов, обеспечивающих конфиденциальность данных [105]. Например, некоторые технологии улучшенной конфиденциальности [135], такие как многослойные вычисления, которые могут быть использованы для предоставления защиты некоторым протоколам [107].

### **2.6.3.5 Отказоустойчивость**

Ожидается, что популяция электронных потребительских устройств будет исчисляться миллиардами, которые служат источниками данных и информации. Такие устройства могут перестать работать или работать некорректно, но они также могут производить неверные или подмененные данные. Поэтому в контексте Интернета вещей необходимо рассматривать проблемы обеспечения отказоустойчивости.

В случае если одно из устройств перестало работать, то необходимо найти устройство, которое может предоставить такие же данные. Для распределенных сетей в таких случаях необходимо разрабатывать механизмы обнаружения ресурсов.

На 2012 год почти не проводилось анализов механизмов, которые могли бы быть использованы для обеспечения жизнеспособности Интернета вещей. Хотя все же существуют разные исследовательские подходы, которые могут быть использованы как основа для реализации отказоустойчивости. Например, инструменты, позволяющие человеку создать ментальную модель окружения, также могли бы использоваться для обнаружения некорректно работающих устройств. Также существуют различные теоретические подходы, нацеленные на реализацию механизмов поиска, обнаружения в Интернете вещей [108].

### **3 Разработка программно-аппаратного стенда с подсистемой виртуальных серверов, подсистемой интеграции больших массивов гетерогенных данных и симуляцией распределенных электронных потребительских устройств (Internet of Things)**

#### **3.1 Назначение программно-аппаратного стенда**

Программно-аппаратный стенд (далее - ПАС) предназначен для моделирования распределенной сети электронных потребительских устройств в целях проведения экспериментальных исследований экспериментального образца масштабируемой сервис-ориентированной программно-аппаратной платформы для сбора, нормализации, обработки и визуализации больших массивов гетерогенных (структурированных, полуструктурированных и неструктурированных) данных в распределенной сети электронных потребительских устройств.

#### **3.2 Состав программно-аппаратного стенда**

В состав программно-аппаратного стенда входят следующие части:

- a) аппаратное обеспечение
  - 1) блейд-сервер, предназначенный для работы ПАС
- b) программное обеспечение
  - 1) подсистема виртуальных серверов
  - 2) подсистема хранения текущих моделируемых показаний электронных потребительских устройств
  - 3) подсистема симуляции распределенных электронных потребительских устройств (Internet of Things)
  - 4) подсистема интеграции больших массивов гетерогенных данных

### **3.3 Аппаратное обеспечение**

#### **3.3.1 Процессор**

Согласно техническому заданию процессор ПАС должен быть не хуже Intel Xeon 5600/5500. Поэтому, в соответствии с таблицей 3.1, были выбраны три процессора того же семейства, характеристики, которых не хуже требуемого. Далее на основе тестов компании PassMark Software мы видим, что процессор Intel Xeon E3-1280 является подходящим вариантом для ПАС.

Процессор Intel Xeon E3-1280V2 Quad-Core, 3.60ГГц, LGA1155, 5.0 GT/s, 8М, OEM - четырехъядерный процессор, для специализированного компьютера. Данный процессор обеспечивает требуемую производительность, необходимую для работы программно-аппаратного стенда. Процессор выполнен с соблюдением норм 22 нанометрового технического процесса.

Основные характеристики процессора, в соответствии с Таблица 3.2: тактовая частота 3.6 ГГц, кэш-память общим объемом 8 МБ, ориентация под платформу LGA1155 и требование к подсистеме питания процессора PCG 05A. Системная шина, на которую рассчитан процессор и её частота довольно велика – 5,00 GT/s. Процессор изготовлен по технологии Intel Viiv и Intel vPro и поддерживает 64-разрядную архитектуру ОС.

Таблица 3.1 – Таблица сравнения процессоров Intel Xeon

Процессор	Тест PassMark (больше - лучше)	Рейтинг (меньше - лучше)	Соотношение цена / качество (производительность) процессора (больше - лучше)
Intel Xeon E3-1245 v3 @ 3.40GHz	9151	50	NA
Intel Xeon X5690 @ 3.47GHz	9258	47	5.36
Intel Xeon E3-1280 V2 @ 3.60GHz	9843	31	14.91

Таблица 3.2 – Спецификация процессора

Наименование	Описание
Характеристики	Процессор Ivy Bridge-H2 Quad Core 3.6GHz (LGA1155,8MB,DMI,69Bt,22nm) Tray Xeon E3-1280 V2
Основные характеристики	
Частота процессора	3.6 ГГц
Модельный ряд	Intel Xeon
Модель	E3-1280V2
Ядро	Ivy Bridge-H2
Количество ядер	4
Сокет	LGA1155
Поддерживается чипсетами	Intel B75, Intel C202, Intel C204, Intel C206, Intel C216, Intel H61, Intel H67, Intel H77, Intel Q65, Intel Q67, Intel Q75, Intel Q77, Intel Z75, Intel Z77
Поддержка Hyper-Threading	Есть
Поддержка технологии Turbo Boost	Есть
Частота процессора в режиме Turbo Boost	4 ГГц
Техпроцесс	22 нм
Базовая частота	100 МГц
Значение коэффициента умножения	36
Редакция PCI-Express	3.0
Количество линий PCI-Express	20
Параметры кэш-памяти	
Объем кэша L1	256 кБ
Объем кэша L2	1 МБ
Объем кэша L3	8 МБ
DMI (Direct Media Interface)	Есть
Максимальная пропускная способность процессорной шины	5 ГТ/с
Поддерживаемые технологии	
Поддержка AMD64/Intel 64/EM64T	Есть
Поддерживаемые инструкции	AVX, MMX, SSE, SSE2, SSE3, SSE4
Поддержка NX Bit	Есть
Поддержка Virtualization Technology (VT-x)	Есть

## Продолжение таблицы 3.2

Наименование	Описание
Характеристики поддерживаемой памяти	
Максимальный объем поддерживаемой памяти	32 ГБ
Максимальная поддерживаемая частота	1600 МГц
Минимальная поддерживаемая частота	1333 МГц
Тип памяти	DDR3
Количество поддерживаемых каналов	2
Максимальная пропускная способность шины памяти	25.6 ГБ/с

### 3.3.2 Материнская плата

Выбор материнской платы зависит от различных характеристик выбранного процессора, такие как поддерживаемые чипсеты, сокет и т.д. На основе этих характеристик была выбрана материнская плата Intel S1200BTSR Beartooth, Socket-LGA1155, iC202, 4xDDR3, 6xSATA, Video, 2xGbLAN, mATX – серверная системная плата Intel, в соответствии с таблицей 3.3.

Таблица 3.3 – Спецификация материнской платы

Наименование	Описание
Тип разъема процессора	Socket 1155
Кол-во устанавливаемых процессоров	1
Поддерживаемые процессоры	Intel Xeon E3-1200 series, Intel Core i3-2100 series
Форм фактор	Micro ATX
Чипсет	Intel® C202
Тип модулей памяти	DDR3 ECC UDIMM
Максимальный объем памяти, ГБ	32
Частота памяти	1066 МГц 1333 МГц
Встроенное видео	Есть
Аудио	Нет
Сеть	10/100/1000
Количество сетевых карт	2
RAID	Есть, Intel® Rapid Storage RAID through onboard SATA connectors provides SATA RAID 0, 1, 5, and 10.

### Продолжение таблицы 3.3

Наименование	Описание
Разъемы для плат расширения	
PCI Express x8	2, 1 слот PCI Express 2.0 в формате x16 1 слот PCI Express 2.0 в формате x8
PCI Express x4	1, 1 слот PCI Express 2.0 в формате x8
PCI	1
Порты и разъёмы	
Внутренние порты и разъемы	6 x SATA II 1 x Chassis intrusion 1 x TPM header 1 x 24-pin ATX power connector 1 x 8-pin ATX 12V power connector 1 x SATA_SGPIO 4 x 4-pin fan headers 1 x SATA Key 3 x USB 2.0 1 x Front Panel Control Header
Задняя панель	1 x VGA 2 x RJ-45 LAN 4 x USB 2.0 1 x Serial
Дополнительные параметры	
Особенности	Встроенное видео: Silicon Motion SM712GX04LF02-BA Встроенная сеть: One Gigabit Ethernet device 82574L connect to PCI-E x1 interfaces on the PCH. One Gigabit Ethernet PHY 82579 connected to PCH through PCI-E x1 interface Поддержка памяти: 1066/1333 MHz ECC Unbuffered (UDIMM) DDR3

#### 3.3.3 Оперативная память

На основе поддерживаемых характеристик процессора и материнской платы для ПАС была выбрана оперативная память Kingston, PC3-10667 1333MHz DDR3 ECC CL9 1Rx8 DIMM, KVR13LE9S8/4 - модуль небуферизированной памяти типа DDR3, объёмом 4 ГБ, с частотой 1333 МГц и пропускной способностью 10600

Мб/сек, в соответствии с таблицей 3.4. Данная оперативная память является высококачественным вариантом для программно-аппаратного стенда.

Таблица 3.4 – Спецификация оперативной памяти

Назначение	Серверная
Производитель	Kingston
Модель	KVR13LE9S8/4
Характеристики	
Объем	4 ГБ
Тип памяти	DDR3 ECC
Тип исполнения	DIMM
Рабочая частота (пропускная способность)	1333 МГц (PC-10600)
Тайминги	CL9
Номинальное напряжение, В	1,35

### 3.3.4 Жесткий диск

Так же на основе поддерживаемых характеристик материнской платы и требований ТЗ для ПАС был выбран жесткий диск Samsung Barracuda SpinPoint F3 ST1000DM005 HD103SJ. Этот диск из линейки SpinPoint F3 емкостью 1 терабайт с интерфейсом SATA 3.0 и скоростью вращения шпинделя 7200об./мин, в соответствии с таблицей 3.5. Данный диск является оптимальным вариантом с его комбинацией надёжности и высокой производительности. Данный жесткий диск подходит для мощной производительной системы, а также может использоваться для сетевых хранилищ данных и для файловых серверов.

Таблица 3.5 – Спецификация жесткого диска

Название	Описание
Производитель	Samsung
Модель	HD103SJ 1 Тб
Параметры производительности	
Время включения	11 секунд - типичное
Скорость вращения шпинделя	7200 оборотов/мин.
Буфер HDD	32 Мб
Среднее время доступа	8.9 мс - типичное
Среднее время ожидания	4.17 мс

## Продолжение таблицы 3.5

Интерфейс, разъемы и выходы	
Поддержка NCQ	Есть
Интерфейс <u>HDD</u>	SATA-II
Пропускная способность интерфейса	300 Мб/сек
Питание	
Потребление энергии в режиме Idle	6.4 Вт
Потребление энергии при поиске	6.9 Вт
Потребление энергии при чтении/записи	7.4 Вт
Потребление энергии в режимах Standby и Sleep	1 Вт - типичный в режиме ожидания

### 3.3.5 Соответствие аппаратной части требованиям технического задания

В соответствии с таблицей 3.6, все требования технического задания, на аппаратную часть разработанного ПАС были соблюдены.

Таблица 3.6 – Сравнение характеристик аппаратного обеспечения с ТЗ

Наименование	Требования к серверу ПАС по ТЗ	Реализация в ПАС	
		кол-во	название
оперативная память GByte/CPU	не менее 2;	4	Kingston, PC3-10667 1333MHz DDR3 ECC CL9 1Rx8 DIMM, KVR13LE9S8/4
дисковое пространство GByte/CPU;	не менее 20	1	HDD 1 Tb Samsung Barracuda SpinPoint F3 ST1000DM005 HD103SJ 3.5" 7200rpm 32Mb
модель процессора	не хуже Intel Xeon 5600/5500	1	Intel Xeon E3-1280V2 Quad-Core, 3.60ГГц, LGA1155, 5.0 GT/s, 8М, OEM
материнская плата	нет требований	1	Intel S1200BTSR Beartooth, Socket- LGA1155, iC202, 4xDDR3, 6xSATA, Video, 2xGbLAN, mATX

## **3.4 Программное обеспечение**

### **3.4.1 Архитектура программного обеспечения ПАС**

В соответствии с рисунком 3.1, программное обеспечение ПАС работает на платформе виртуализации, которая обеспечивает масштабируемость ПАС через увеличение количества виртуальных машин. Виртуальная машина состоит из следующих частей:

- а) Подсистема виртуализации, которая обеспечивает необходимую программную среду для работы ПАС
- б) Подсистема хранения текущих показаний моделируемых электронных потребительских устройств
- в) Подсистема симуляции распределенных электронных потребительских устройств. Данная подсистема считывает конфигурационные настройки из БД и записывает текущие моделируемые показания в БД
- г) Подсистема интеграции больших массивов гетерогенных данных. Данная подсистема считывает текущие моделируемые показания из БД и реализует REST подобный интерфейс с помощью протокола СоАР, через который внешние системы получают доступ к моделируемым показаниям электронных потребительских устройств.

Подсистема хранения, подсистема симуляции и подсистема интеграции размещаются на одной или разных виртуальных машинах. Также подсистема симуляции и подсистема интеграции выполняются в программной среде платформы Java Virtual Machine.

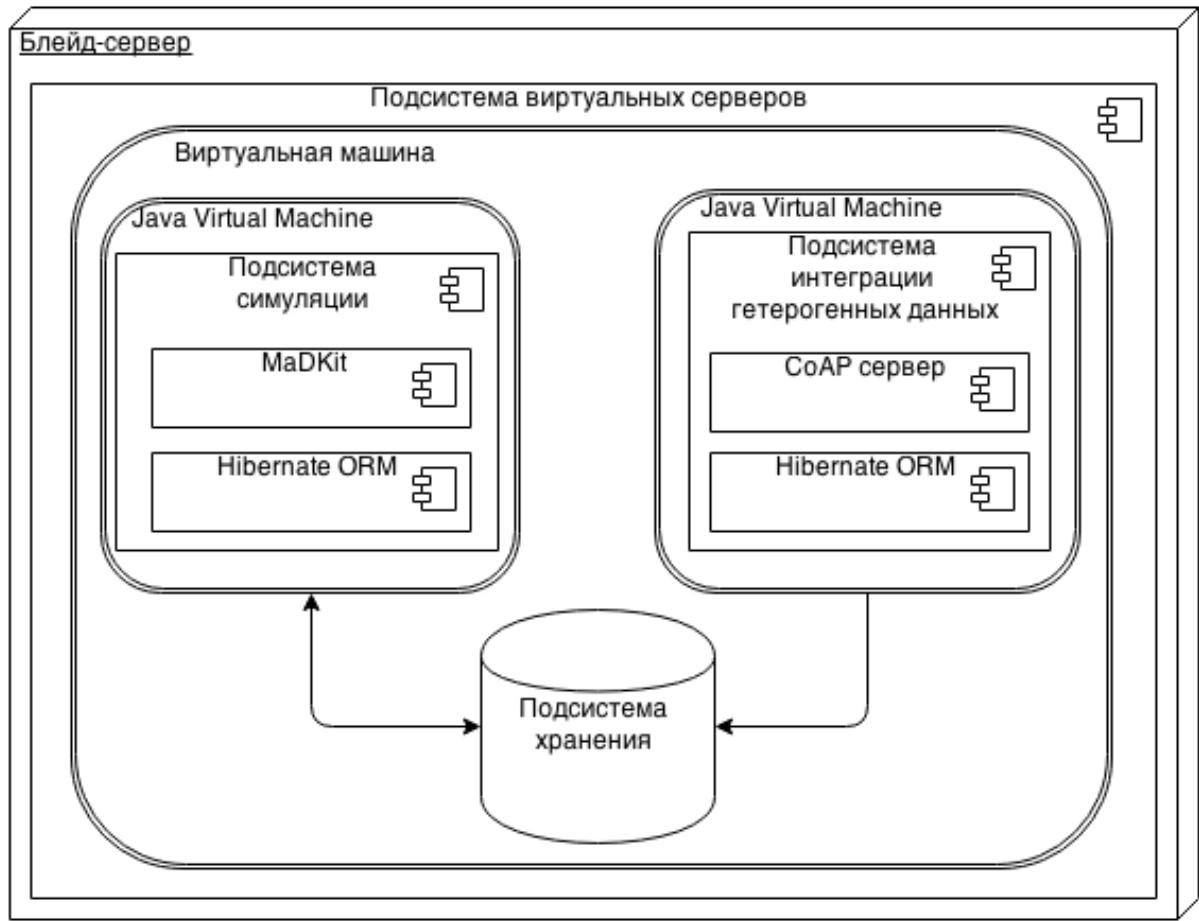


Рисунок 3.1 – Архитектура программного обеспечения ПАС

### 3.4.2 Подсистема виртуальных серверов

#### 3.4.2.1 Назначение и роль подсистемы виртуальных серверов

Подсистема предназначена для обеспечения необходимой программной среды, которая позволяет безопасно для всего ПАС выполнять моделирование различных сценариев работы облака сенсоров и шлюзов. Подсистема также позволяет испытать работу системы при максимальной нагрузке в различных конфигурациях оборудования, задаваемого параметрами виртуальных машин.

Подсистема виртуализации имеет сервисную, инфраструктурную роль, но именно она на базовом уровне является условием работы ПАС в целом.

Основным преимуществом применения системы виртуальных серверов является возможность обеспечить симуляцию разнородной топологической сети при высокой гибкости настройки и относительно не высокой цене системы.

### **3.4.2.2       Функции подсистемы виртуальных серверов**

К основным функциям подсистемы следующие:

- а) Обеспечить необходимую гибкость и вариативность конфигураций аппаратного обеспечения при нагрузочном и прочих видах тестирования\
- б) Предоставить возможность тестировать различные топологии облаков сенсоров и конфигураций шлюзов

### **3.4.2.3       Характеристика наиболее распространенных платформ виртуализации**

#### **3.4.2.3.1     vSphere**

VMware vSphere — это ведущая в отрасли платформа виртуализации для создания облачных инфраструктур.

Решение VMware® vSphere® предназначено для организаций, которые планируют виртуализировать свои ЦОД целиком и перейти к модели «ИТ как услуга». Это решение включает возможности для преобразования ЦОД в упрощенные облачные среды, в которых можно предоставлять гибкие и надежные ИТ-услуги следующего поколения.

- а) vSphere помогает ИТ-отделам выполнять соглашения об уровне обслуживания для самых важных и ресурсоемких приложений и обеспечивает самую низкую совокупную стоимость владения.
- б) vSphere ускоряет перенос существующих ЦОД в вычислительное облако и обеспечивает подключение к совместимым общедоступным облакам, формируя основу для уникальной в отрасли гибридной облачной модели.
- в) Поддержка более 3 000 приложений, разработанных более чем 2 000 независимыми партнерами-поставщиками ПО, делает VMware vSphere самой надежной платформой для любых приложений.

## Использование vSphere

- а) Доступность и производительность: улучшенная доступность и производительность для важных приложений и приложений следующего поколения (например, Hadoop).
- б) Хранилище: кэширование на стороне сервера для повышения производительности приложений.
- в) Масштабируемость: поддержка даже самых ресурсоемких рабочих нагрузок благодаря удвоению максимальных параметров конфигурации в некоторых основных областях. Основные возможности и компоненты vSphere Платформа виртуализации
- г) Архитектура гипервизора VMware vSphere предоставляет надежный, проверенный в производственных средах и высокопроизводительный уровень виртуализации. С его помощью несколько виртуальных машин могут совместно использовать аппаратные ресурсы и обеспечивать производительность, сопоставимую со стандартной производительностью систем (а в некоторых случаях превышающую ее).
- д) Технология виртуальной симметричной многопроцессорной обработки VMware vSphere (Virtual SMP) дает возможность использовать сверхмощные виртуальные машины, содержащие до 64 виртуальных ЦП.
- е) Файловая система vSphere VMFS 5 предоставляет виртуальным машинам доступ к общим устройствам хранения (Fibre Channel, iSCSI и др.) и является основной технологией для других компонентов vSphere, таких как VMware vSphere Storage vMotion®.
- ж) API-интерфейсы хранилища VMware vSphere обеспечивают интеграцию с поддерживаемыми сторонними решениями по защите данных, передаче по нескольким путям ввода-вывода и организации работы дисковых массивов.

Основные преимущества:

- а) Эффективность за счет высоких коэффициентов использования и автоматизации: достижение коэффициента консолидации 15:1 и увеличение коэффициентов использования оборудования с 5–15% до 80% и более без ущерба для производительности.
- б) Значительное снижение ИТ-расходов: сокращение капитальных расходов на 70%, а эксплуатационных расходов на 30% обеспечивает снижение общих расходов на инфраструктуру на 20–30% из расчета на приложение в среде vSphere.
- в) Адаптивность с сохранением контроля: оперативное реагирование на изменяющиеся потребности бизнеса без ущерба для безопасности или контроля, реализация автономной и самоуправляемой инфраструктуры со встроенными средствами обеспечения доступности, масштабируемости и производительности всех важных приложений, работающих в среде vSphere.
- г) Свобода выбора: общая платформа на основе единых стандартов обеспечивает использование существующих ИТ-ресурсов и предоставление ИТ-услуг следующего поколения, а также расширяет возможности vSphere с помощью открытых API-интерфейсов, реализующих взаимодействие с решениями ведущих поставщиков технологий, объединенных в глобальную экосистему

Недостатки:

- а) Высокая стоимость лицензий;
- б) Высокие требования к аппаратному обеспечению

#### 3.4.2.3.2 OpenVZ

Одной из широко распространенных реализаций виртуальных машин является OpenVZ. OpenVZ реализует технологии виртуализации на уровне операционной системы и базируется на ядре Linux. OpenVZ позволяет на одном физическом

сервере запускать множество изолированных копий операционной системы, называемых виртуальные частные серверы или виртуальные среды.

OpenVZ является базовой платформой для Virtuozzo — проприетарного продукта Parallels, Inc. OpenVZ распространяется на условиях лицензии GNU GPL v.2.

Поскольку OpenVZ базируется на ядре Linux, в отличие от виртуальных машин (напр. VMware, Parallels Desktop) или паравиртуализационных технологий (напр. Xen), в роли «гостевых» систем могут выступать только дистрибутивы Linux.

Виртуализация на уровне операционной системы в OpenVZ даёт лучшую производительность, масштабируемость, плотность размещения, динамическое управление ресурсами, а также лёгкость в администрировании, чем у альтернативных решений. Согласно сайту OpenVZ, накладные расходы на виртуализацию очень малы, и падение производительности составляет всего 1-3 %, по сравнению с обычными Linux-системами.

OpenVZ состоит из модифицированного ядра Linux и пользовательских утилит.

Ядро OpenVZ — это модифицированное ядро Linux, добавляющее концепцию виртуальной среды. Ядро обеспечивает виртуализацию, изоляцию, управление ресурсами и сохранением текущего состояния виртуальной среды.

Каждая VE — это отдельная сущность, и с точки зрения владельца VE она выглядит практически как обычный физический сервер. Каждая VE имеет свои собственные:

- a) Файлы: Системные библиотеки, приложения, виртуализованные ФС /proc и /sys, виртуализованные блокировки и т. п.
- б) Пользователи и группы: Свои собственные пользователи и группы, включая root.

- в) Дерево процессов: VE видит только свои собственные процессы (начиная с init).
- г) Сеть: Виртуальное сетевое устройство (venet), позволяющая VE иметь свои собственные адреса IP, а также наборы правил маршрутизации и файрволла
- д) Устройства: При необходимости администратор OpenVZ сервера может дать VE доступ к реальным устройствам, напр. сетевым адаптерам, портам, разделам диска и т. д.
- е) Объекты IPC: Разделяемая память, семафоры, сообщения.

OpenVZ использует одно ядро для всех VE, поэтому система столь же масштабируема, как обычное ядро Linux 2.6, то есть поддерживает максимально до 4096 процессоров и до 64 ГБ оперативной памяти для x86-версии (с использованием PAE) и 64ТБ для x86-64.

Единственная виртуальная среда может быть расширена до размеров всего физического сервера, то есть, использовать все доступное процессорное время и память.

При этом OpenVZ способна размещать сотни виртуальных сред на одном физическом сервере при современном аппаратном обеспечении. Основными ограничивающими факторами являются объём ОЗУ и частота процессора. Из экспериментальных данных, на машине с 2 GB памяти возможно запустить примерно 300-320 VE при условии сохранения хорошей производительности.

Владелец физического сервера с OpenVZ (root) может видеть все процессы и файлы всех VE. Эта особенность делает возможным массовое управление, в отличие от других технологий виртуализации (напр. VMware или Xen), где виртуальные сервера являются отдельными сущностями, которыми невозможно напрямую управлять с хост-системы.

### 3.4.2.3.3 Xen

Xen – кроссплатформенный гипервизор, разработанный в компьютерной лаборатории Кембриджского университета и распространяемый на условиях лицензии GPL. Основные особенности: поддержка режима паравиртуализации помимо аппаратной виртуализации, минимальность кода самого гипервизора за счёт выноса максимального количества компонентов за пределы гипервизора.

Xen обладает функциональностью ПО корпоративного уровня; в нём, в частности, обеспечивается:

- а) Производительность виртуальных машин близкая к производительности при непосредственном исполнении на железе;
- б) Возможность живой миграции работающих виртуальных машин между хостами;
- в) Поддержка до 32 виртуальных процессоров на одну гостевую машину с возможностью горячего добавления процессоров;
- г) Поддержка платформ x86/32, x86/32 с PAE, x86/64, IA64, а также частичная поддержка платформ ARM и PPC;
- д) Поддержка аппаратной виртуализации для запуска не модифицированных операционных систем (включая Microsoft Windows);
- е) Широкая поддержка оборудования (поддерживаются практически все драйверы устройств Linux).

Xen начался как исследовательский проект Кембриджского университета под руководством Иэна Прэтта, ставшего в дальнейшем основателем компании XenSource. Первый публичный релиз Xen выпущен в 2003 году. В октябре 2007 Citrix купила XenSource и осуществила переименование продуктов.

21 октября 2009 Citrix объявила, что коммерческие версии XenServer станут полностью свободными. Хотя существует свободная версия Citrix XenServer, для XenCenter (программного обеспечения для централизованного управления) не предоставляется исходных кодов, хотя она доступна бесплатно для загрузки.

В Xen основной концепции гипервизора является домен. Под доменом понимается запущенная копия виртуальной машины. Соответственно, Xen оперирует только понятием домена, а понятие «виртуальной машины» появляется на уровне администрирования (прикладных программ, управляющих гипервизором). Если виртуальная машина перезагружается, то её домен завершается (в момент перезагрузки) и появляется новый домен. Более того, даже при миграции содержимое копируется из одного домена в другой домен. Таким образом, за время своей жизни практически все виртуальные машины оказываются по очереди в разных доменах.

Традиционно используется следующая терминология:

- а) frontend — часть модуля, находящегося в domU,
- б) backend — часть, находящаяся в dom0.

Для некоторых типов устройств backend часть может быть различной при сохранении одной и той же frontend части. Например, драйвер блочного устройства может иметь backend в форме программы работы с VHD-образами, с блочными устройствами, с iSCSI-инициатором и т. д.

Выделяют следующие типы доменов:

- а) dom0 – базовый домен Xen. Обычно он автоматически создаётся и загружается сразу после загрузки и инициализации гипервизора. Этот домен имеет особые права на управление гипервизором и по умолчанию всё аппаратное обеспечение компьютера доступно из dom0. dom0 всегда один.
- б) domU – рядовой домен (сокращение от User domain), содержащий в себе домен выполняющихся виртуальных машин. Обычно не имеет доступа к реальному оборудованию и является «полезной нагрузкой» системы виртуализации. В отличие от dom0, domU может быть множество (несколько десятков).

- в) stub-domain – домен, в котором запущена специализированная ОС, обеспечивающая работу с каким-либо оборудованием или бэк-эндом драйвера. Является развитием модели безопасности Xen.
- г) domain builder (конструктор доменов) – программа, которая создаёт domU (загружает в него нужный код и сообщает гипервизору о необходимости запуска). Помимо конструирования домена, обычно занимается подключением и конфигурированием виртуальных устройств, доступных для виртуальной машины. Она же отвечает за процесс миграции виртуальной машины с хоста на хост.

Паравиртуализация (PV) требует адаптации ядра исполняемой ОС для работы совместно с Xen (см. выше). Позволяет достичь очень высокой производительности за счёт отсутствия эмуляции «настоящего железа», простоты интерфейсов и учёта существования гипервизора при выполнении системных вызовов в коде ядра.

При паравиртуализации выполнение привилегированных операций запрещено, вместо них совершаются гипервизором обращения ядра гостевой ОС к гипервизору с просьбой о выполнении тех или иных операций.

В большинстве случаев изменения при портировании ОС под Xen затрагивают только ядро ОС, хотя могут требоваться и незначительные изменения в системных библиотеках.

Так как драйверы в Xen состоят из двух частей, одна из которых исполняется вне виртуальной машины, а вторая находится внутри неё – в гостевой системе. Эта часть крайне примитивна и служит лишь транслятором запросов в часть, принадлежащую гипервизору.

В PV-режиме не поддерживаются «вложенные» режимы работы процессора, такие как real-86, virtual-86, переключение между 32-битным и 64-битным режимом, поддержка эмуляции аппаратной виртуализации и т. д. В связи с этим в PV-режиме отсутствует начальный фрагмент загрузки компьютера (с имитацией кода BIOS,

загрузчика и т. д.), а ядро гостевой системы сразу же запускается в нужном режиме, подобно тому, как запускаются обычные программы. В связи с этим, в частности, сам Xen не может работать в PV-режиме (то есть невозможно запустить «вложенный» гипервизор в PV-режиме).

В режиме аппаратной виртуализации (HVM) гостевая ОС не адаптируется, не «знает» про существование гипервизора. Xen с помощью модулей из QEMU эмулирует реальное аппаратное обеспечение и позволяет провести начальную загрузку ОС. По её окончании для нормальной производительности должны запускаться PV-драйверы, которые реализуют быстрый интерфейс с виртуальными устройствами, подобно тому, как это работает в PV-режиме). Поскольку большинство привилегированных операций эмулируется, возможен запуск Xen в HVM-режиме из-под Xen. В этом случае вложенный гипервизор сможет работать только в PV-режиме.

Гипервизор Xen реализует минимальный набор операций для управления оперативной памятью, состоянием процессора, таймерами реального времени и счётчиками тактов процессора, прерываниями и контролем за DMA. Все остальные функции, такие как реализация дисковых и блочных устройств, создание и удаление виртуальных машин, их миграция между серверами и т. д. реализуется в управляющем домене. За счёт этого размер гипервизора получается весьма малым (для версии 3.4 размер двоичного кода всего гипервизора меньше 600 КБ).

Программа в dom0 выполняет основную часть работы. При этом программа чаще всего запускается в виде отдельного процесса для каждого обслуживаемого устройства. Сбой в такой программе ведёт к сбою только одного устройства (блочного, сетевого) и не затрагивает работу других копий программы (то есть не затрагивает сетевые/блочные устройства остальных доменов, или даже другие устройства того же самого домена).

Все функции, связанные с обеспечением работы сети, блочных (дисковых) устройств, эмуляции видеоадаптеров и прочих устройств вынесены за пределы

гипервизора. Большинство таких устройств состоит из двух частей: драйверы в domU и программы в dom0. Драйвер, встроенный в ядро ОС или загружающийся в виде модуля, реализует минимальный объём работы, фактически, транслируя запросы от ОС в программу в dom0.

Это увеличивает устойчивость системы виртуализации, так как ошибка в компонентах вне гипервизора не приводит к повреждению самого гипервизора и ограничивает повреждения только вышедшим из строя компонентом, не мешая работать остальным.

Xen предоставляет доменам три механизма взаимодействия: один — с гипервизором, и два между доменами. Чаще всего, взаимодействие происходит между dom0 и domU, хотя модель допускает взаимодействие и между двумя domU.

Xen (с помощью стека управления) поддерживает миграцию гостевых виртуальных машин по сети. Миграция паравиртуальных машин поддерживается с версии Xen 2, а HVM — с версии 3. Миграция может происходить как с выключением гостевой системы, так и прямо в процессе работы, так называемая «живая» миграция без потери доступности.

Необходимо, чтобы оба физические сервера Xen видели одно и то же хранилище, на котором находятся данные виртуальной машины. Это требуется потому, что при миграции виртуальной машины её файловая система не копируется, так как это требовало бы слишком много времени даже в случае быстрой сети. Общее хранилище может быть организовано на основе различных технологий SAN или NAS, например Fibre Channel, iSCSI или DRBD.

На базе Xen могут быть созданы следующие системы:

- а) Хост-системы
- б) Гостевые системы
- в) Облачные системы

Xen широко применяется как компонент виртуализации в облачных вычислениях и при организации служб выделенных частных серверов. Такие хостинговые компании как Amazon Elastic Compute Cloud, Liquid Web, Fujitsu Global Cloud Platform, Linode, SparkNode и Rackspace Cloud используют Xen как гипервизор виртуальных машин.

В настоящее время сообщество Xen разрабатывает Xen Cloud Platform (XCP) — систему серверной виртуализации. Свое происхождение XCP ведет от бесплатной версии Citrix XenServer и выпускается полностью под GNU GPL.

На основе Xen создано несколько коммерческих продуктов для консолидации серверов. В частности это такие продукты как:

Citrix XenServer — коммерческий продукт, имеется также бесплатная версия XenServer, в которой отключены некоторые функции. Стек управления — на основе xapi, имеется внешняя программа управления Citrix XenCenter.

Virtual Iron - С 2009 года права на продукт принадлежат корпорации Oracle, с июня 2009 года продукт не развивается и не поставляется заказчикам.

Sun xVM — первоначально Sun xVM hypervisor и Sun xVM Server — на основе кода гипервизора Xen для OpenSolaris. С мая 2009 года разработка xVM Server ведется в рамках проекта Xen/OpenSolaris.

Red Hat Enterprise Linux версии 5 (RHEL5) включает в себя опцию Virtualization на основе гипервизора Xen и стека управления xend. Для управления используются virt-manager или libvirt/virsh. В RHEL6 для виртуализации Xen не используется, вместо него применяется KVM, но имеется поддержка Xen в domU.

Oracle VM — Продукт включает Oracle VM Server for x86 и Oracle VM Manager. Сервер и менеджер бесплатны и доступны для скачивания. Стек управления на основе xend. Для управления сервером на него устанавливается специальный Oracle VM Agent (ovs-agent). Manager представляет собой клиент-

серверное приложение на основе WebLogic и позволяет управлять серверами виртуализации из браузера.

Oracle Linux подобно Red Hat Enterprise Linux содержит виртуализацию на основе Xen для версии 5 и KVM для версии 6.

SUSE Linux Enterprise Server версий SLES10 и SLES11 поддерживают Xen или KVM на выбор.

#### 3.4.2.3.4 KVM

Не менее широко распространенной реализацией виртуальных машин является KVM (Kernel-based Virtual Machine) — это программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) или AMD SVM (Secure Virtual Machine).

Программное обеспечение KVM состоит из:

- а) загружаемого модуля ядра (называемого kvm.ko), предоставляющего базовый сервис виртуализации,
- б) процессорно-специфического загружаемого модуля kvm-amd.ko либо kvm-intel.ko,
- в) и компонентов пользовательского режима (модифицированного QEMU).

Все компоненты ПО KVM являются ПО с открытым исходным кодом. Компонент ядра, необходимый для работы KVM, включен в основную ветку Linux начиная с версии 2.6.20 (февраль 2007). KVM был также портирован на FreeBSD как модуль ядра. Ведётся работа по включению модификаций, необходимых для работы с KVM, в основную ветку QEMU.

Сам по себе KVM не выполняет эмуляции. Вместо этого программа, работающая в пространстве пользователя, использует интерфейс /dev/kvm для настройки адресного пространства гостя виртуальной машины, через него же эмулирует устройства ввода-вывода и видеоадаптер.

KVM позволяет виртуальным машинам использовать немодифицированные образы дисков QEMU, VMware и других, содержащие операционные системы. Каждая виртуальная машина имеет своё собственное виртуальное аппаратное обеспечение: сетевые карты, диск, видеокарту и т. д.

Программное обеспечение KVM было создано, разрабатывается и поддерживается фирмой Qumranet, которая была куплена RedHat за 107 миллионов долларов 4 сентября 2008 года.

KVM требует наличия x86-совместимого процессора с поддержкой одной из технологий аппаратной виртуализации — Intel VT либо AMD SVM. На данный момент KVM в состоянии запускать в качестве гостевых ОС:

- а) GNU/Linux (32-битные и 64-битные),
- б) Windows (32-битные и 64-битные)
- в) и другие системы.

KVM использует модифицированный QEMU в качестве фронтенда. Ведётся работа по оптимизации использования возможностей аппаратной виртуализации, заложенных в современных процессорах Intel и AMD. Производительность KVM сравнивалась с производительностью Xen, работающего в режиме аппаратной виртуализации (не паравиртуализации), и при определённых видах нагрузок KVM показывал лучшую производительность (но не в большинстве случаев).

KVM в настоящий момент поддерживает команды Migration: savevm/loadvm и оффлайн (offline) или живую миграцию. Команды миграции даются в qemu-мониторе (Alt-Ctrl-2). После успешного завершения перемещенный VM продолжает работать на назначенному хосте.

Вы можете переместить гостя между узлом AMD на хост Intel и назад. Естественно, 64-разрядный гость может быть перемещен только на 64-разрядный узел, но 32-разрядный гость может быть перемещен по желанию.

Встречаются более старые процессоры Intel, которые не поддерживают NX (или XD), которые могут вызвать проблемы в кластере, включающем узлы поддерживающие NX. Чтобы отключить NX для данного гостя, запустите это с такого параметра.

Образ виртуальной машины доступен и на источнике и на конечных хостах (расположенный на совместно используемой памяти, например, используя nfs).

Достоинства:

- а) Эти возможности уникальны для KVM Live Migration на данный момент.
- б) Почти не отмечаемое время простоя гостя
- в) Гость не вовлекается (уникальный для KVM Живая Миграция [#1 1])
- г) Возможность туннелировать статус VM через внешнюю программу (уникально для KVM Живая Миграция [#1 1])
- д) ssh/gzip/bzip2/gpg/your own
- е) При успехе гость продолжает работать на конечном хосте. при ошибке - гость продолжает работать на исходном узле (с одним исключением)
- ж) Короткий и Простой
- з) Легкий для улучшения
- и) Аппаратная независимость (почти).
- к) Поддержка миграции остановленных (приостановленных) VMs.
- л) Открытый

#### **3.4.2.4       Принятое решение о платформе виртуализации**

В соответствии с требованиями ТЗ принято решение остановить выбор на бесплатно распространяемой версии платформы виртуализации VMware vSphere 5.5.

Основными преимуществами выступает то, что она обеспечивает достаточно широкие возможности по созданию виртуальных серверов, в частности,

поддерживает возможность установки в качестве виртуальных ОС ОС Linux Ubuntu Server 14.10 и Windows 7 и выше. При необходимости, возможен переход на проприетарную версию.

Важно отметить, что все недостатки использования vSphere нивелированы в данном проекте его спецификой. В частности, дороговизна лицензии перестает играть роль ввиду наличия бесплатной версии с ограниченной функциональностью, наличие которой не принципиально для целей проекта. Вторым недостатком данного продукта считают его высокие требования к аппаратной платформе. Так как проект предусматривает наличие только одного физического сервера, который может быть сформирован из довольно современных комплектующих, то данный недостаток тоже не является препятствием.

### **3.4.2.5 Краткий обзор операционных систем для подсистем виртуализации**

#### **3.4.2.5.1 MS Windows**

Платформа виртуализации представляет собой контейнер, в котором существуют экземпляры различных операционных систем. В данном разделе целесообразно остановиться на обзоре наиболее распространенных операционных систем.

Microsoft Windows — семейство проприетарных операционных систем корпорации Microsoft, ориентированных на работу пользователя с использованием графического интерфейса. По состоянию на март 2013 года под управлением операционных систем семейства Windows по данным ресурса NetMarketShare (Net Applications) работало около 90% персональных компьютеров.

В настоящее время можно говорить о следующих семействах Windows:

- а) Семейство Windows 9x – вышли из применения
- б) Семейство Windows NT
- в) Семейство ОС для смартфонов

### г) Семейство встраиваемых ОС Windows Embedded

Семейство Windows NT. Только в этом семействе представлены операционные системы для серверов. Начиная с Windows Server 2003 серверные операционные системы называются добавлением суффикса «Server» и года выпуска. Семейство Windows NT относится к операционным системам с вытесняющей многозадачностью. В основу семейства Windows NT положено разделение адресных пространств между процессами. Каждый процесс имеет возможность работать с выделенной ему памятью, но он не имеет прав для записи в память других процессов, драйверов и системного кода. Операционные системы Windows работают на платформах x86, x86-64, IA-64 и ARM. Ранние версии поддерживали DEC Alpha, MIPS, PowerPC и SPARC.

Семейство ОС для смартфонов (Windows CE) – это семейство операционных систем реального времени для мобильных устройств. Поддерживаются процессоры ARM, MIPS, SuperH и x86. В отличие от остальных операционных систем Windows, операционные системы этого семейства продаются только в составе готовых устройств, таких как смартфоны, карманные компьютеры, GPS-навигаторы, MP3-проигрыватели и другие. В настоящее время под термином «Windows CE» понимают только ядро операционной системы.

Семейство ОС Windows Embedded — это семейство операционных систем реального времени для применения в различных встраиваемых системах. ОС этого семейства продаются только в составе готовых устройств. Ядро системы общее с семейством ОС Windows CE и поддерживает процессоры ARM, MIPS, SuperH и x86, но Windows Embedded включает дополнительные функции по встраиванию, среди которых фильтр защиты от записи (EWF и FBWF), загрузка с флеш-памяти, CD-ROM, сети, использование собственной оболочки системы и т. п.

Семейство ОС Windows Embedded используется в банкоматах, медицинских приборах, навигационном оборудовании, «тонких» клиентах, VoIP-терминалах,

медиа проигрывателях, цифровых рамках (альбомах), кассовых терминалах, платёжных терминалах, роботах, игровых автоматах, музыкальных автоматах и т.д.

Пакет Windows включает в себя «стандартные» приложения, такие как браузер (Internet Explorer), почтовый клиент (Outlook Express или Почта Windows), музыкальный и видеопроигрыватель (проигрыватель Windows Media). С помощью технологий COM и OLE их компоненты могут быть использованы в приложениях сторонних производителей.

#### 3.4.2.5.2 Linux

Linux — общее название Unix-подобных операционных систем, основанных на одноимённом ядре. Ядро Linux создаётся и распространяется в соответствии с моделью разработки свободного и открытого программного обеспечения. Комплектации Linux распространяются в основном бесплатно, подавляющее большинство ПО в современных дистрибутивах по-прежнему доступно по свободным лицензиям, за исключением небольшого количества проприетарных компонентов и распространяется в виде различных готовых дистрибутивов, имеющих свой набор прикладных программ и уже настроенных под конкретные нужды пользователя.

Согласно исследованию Goldman Sachs, в целом, рыночная доля Linux среди электронных устройств составляет около 42 %. Linux лидирует на:

- а) рынках смартфонов (Android занимает 85 % рынка),
- б) интернет-серверов (60 %) (Linux порядка 32 %, 64,1 % указаны как доля Unix),
- в) самых мощных суперкомпьютеров (97 %).
- г) Значительная доля рынка, согласно Linux Foundation, принадлежит Linux
- д) в данных центрах и на предприятиях – половина рынка встраиваемых систем,
- е) на рынке нетбуков – около 32 %.

ж) На рынке домашних компьютеров Linux на 3 месте – до 5 %.

Ядро Linux было создано для x86-ПК, но было портировано на множество платформ, включая x86-64, PowerPC и ARM. Linux работает в роутерах, телевизорах и игровых приставках.

Наиболее популярными дистрибутивами являются:

- а) deb-based (Debian, Mint, Ubuntu),
- б) pacman-based (Arch Linux, Chakra, Manjaro),
- в) RPM-based (RedHat, Fedora, Mageia, OpenSUSE),
- г) source-based (Slackware, Gentoo).

Дистрибутивы Linux используются в качестве серверных операционных систем и, отчасти из-за цены, заняли значительную долю этого рынка. Linux является ключевым компонентом популярного комплекса серверного программного обеспечения LAMP (Linux, Apache, MariaDB/MySQL, Perl/PHP/Python), который стал одной из наиболее распространённых платформ для хостинга веб-сайтов.

Дистрибутивы Linux становятся всё более популярными на мейнфреймах в последнее десятилетие, отчасти из-за цены и модели с открытым исходным кодом. В декабре 2009 компьютерный гигант IBM объявил этот рынок основным и сделал упор на продажу мейнфреймов только с Linux.

Также дистрибутивы Linux широко используются в качестве операционной системы суперкомпьютеров: по данным на июнь 2013, более 95 % компьютеров из списка 500 самых мощных работали под управлением различных вариантов Linux. Операционной системой самого мощного современного суперкомпьютера — Tianhe-2 — является Kylin Linux.

Каждый дистрибутив Linux содержит ряд решений для разных задач пользователей, объединённых едиными системами установки, управления и обновления пакетов, настройки и поддержки.

Некоторые из самых распространённых в мире дистрибутивов:

- a) Debian GNU/Linux — один из старейших дистрибутивов, разрабатываемый обширным сообществом разработчиков. Служит основой для создания множества других дистрибутивов. Отличается строгим подходом к включению несвободного ПО.
- б) Ubuntu — дистрибутив, основанный на Debian и быстро завоевавший популярность. Поддерживается сообществом, разрабатывается Canonical Ltd. Основная сборка ориентирована на лёгкость в освоении и использовании, при этом существуют серверная и минимальная сборки.
- в) Linux Mint — дистрибутив, основанный на Ubuntu и полностью с ним совместимый, включающий в себя по умолчанию Java, Adobe Flash и многое другое.
- г) openSUSE — дистрибутив, разрабатываемый сообществом при поддержке компании Novell. Отличается удобством в настройке и обслуживании благодаря использованию утилиты YaST.
- д) Fedora — поддерживается сообществом и корпорацией RedHat, предшествует выпускам коммерческой версии RHEL.

Linux — ядро операционной системы, соответствующее стандартам *POSIX*. Разработка была начата финским студентом Линусом Торвальдсом в 1991 году. В основном код написан на Си с некоторыми расширениями gcc и на ассемблере (с использованием AT&T-синтаксиса GNU Assembler).

Распространяется как свободное программное обеспечение на условиях GNU General Public License, кроме несвободных элементов, особенно драйверов, которые используют прошивки, распространяемые под различными лицензиями. Владельцем торговой марки Linux является Линус, а помогает следить за соблюдением его прав и условий GPL Фонд свободного программного обеспечения.

Ядро Linux поддерживает многозадачность, виртуальную память, динамические библиотеки, отложенную загрузку, производительную систему управления памятью и многие сетевые протоколы.

На сегодняшний день Linux — монолитное ядро с поддержкой загружаемых модулей. Драйверы устройств и расширения ядра обычно запускаются в 0-кольце защиты, с полным доступом к оборудованию. В отличие от обычных монолитных ядер, драйверы устройств легко собираются в виде модулей и загружаются или выгружаются во время работы системы.

Linux на данное время перенесён на очень широкий круг архитектур, запускается на широком спектре оборудования от iPAQ (карманный компьютер) до IBM S/390 (высокопроизводительный мейнфрейм). Системы на основе Linux используются в качестве основных почти на всех суперкомпьютерах (более 91 % списка Top500), в том числе и на самых мощных — K computer.

Изначально Linux разрабатывался для 32-битных x86-совместимых ПК; на сегодняшний день различные версии ядра Linux запускаются на следующих процессорных архитектурах:

a) ARM:

- 1) Acorn: Archimedes, A5000, RiscPC;
- 2) StrongARM, Intel XScale и тому подобных;
- 3) Axis Communications [CRIS](#);
- 4) DEC Alpha;
- 5) HP PA-RISC;
- 6) Hitachi: SuperH (SEGA Dreamcast), H8/300;
- 7) IBM System/390;
- 8) IBM zSeries-мейнфреймы;

б) Intel 80386 и выше: IBM PC и совместимые с процессорами:

- 1) 80386, 80486, а также AMD, Cyrix, TI и IBM-варианты;
- 2) серия Pentium;
- 3) Core, Core2 Duo в 32 и 64-х битных версиях;
- 4) AMD Am5x86, K5, K6, Athlon (все 32-битные версии), Duron;

в) AMD64: 64-битная технология AMD (также известная как x86-64);

- 1) Cyrix 5x86, 6x86 (M1), 6x86MX и MediaGX (National/AMD Geode) серия;
  - 2) VIA C3 и последующие процессоры;
  - 3) поддержка Intel 8086, 8088, 80186, 80188 и 80286 процессоров находится в разработке (см. проект ELKS (англ.));
  - 4) Microsoft Xbox (Pentium III);
  - 5) Intel IA-64;
- г) MIPS:
- 1) Silicon Graphics, Inc.;
  - 2) Cobalt Qube, Cobalt RaQ;
  - 3) Sony/Toshiba/IBM — Emotion Engine и Cell, используемые в PlayStation 2 и PlayStation 3 соответственно;
  - 4) DECstation
  - 5) и некоторые другие;
- д) Motorola 68020 и выше:
- 1) более новые Amiga: A1200, A2500, A3000, A4000;
  - 2) Apple Macintosh II, LC, Quadra, Centris и ранняя серия Performa;
  - 3) рабочие станции Sun Microsystems серии 3 (экспериментальная, с использованием Sun-3 MMU);
  - 4) NEC v850e;
  - 5) Renesas M32R;
- е) PowerPC и IBM POWER:
- 1) все новые компьютеры Apple (все оснащённые PCI Power Macintoshes, ограниченная поддержка NuBus Power Macs),
  - 2) клоны PCI Power Mac, разработанные Power Computing, UMAX и Motorola;
  - 3) IBM RS/6000, iSeries- и pSeries-системы;
  - 4) Pegasos I и II системы;
  - 5) некоторые встроенные системы PowerPC;
  - 6) Qualcomm Hexagon

7) SPARC и UltraSPARC: Sun 4-series, SPARCstation/SPARCserver, Ultra-, Blade- и Fire-серии рабочих станций и серверов.

### 3.4.2.5.3 FreeBSD

FreeBSD – это современная операционная система для серверов, десктопов и встроенных компьютерных платформ. Основана на BSD UNIX. Её код прошёл через более чем тридцать лет непрерывного процесса развития, совершенствования и оптимизации.

FreeBSD разрабатывается и поддерживается большой командой разработчиков.

FreeBSD обеспечивает современные сетевые возможности, безопасность и производительность на мировом уровне и используется на самых загруженных веб-сайтах мира и на наиболее распространенных встроенных сетевых устройствах и устройствах хранения.

FreeBSD является идеальной платформой для построения Internet или Intranet сервера. Эта система предоставляет надёжные даже при самой интенсивной нагрузке сетевые службы, и эффективное управление памятью, что позволяет обеспечивать приемлемое время отклика для тысяч одновременно работающих пользовательских задач.

OpenBSD — это современная свободная многоплатформенная операционная система, основанная на 4.4BSD-реализации UNIX-системы, вторая по популярности BSD-система

OpenBSD — самостоятельный проект, ответвление NetBSD, возникшее в конце 1995 года. Основное отличие OpenBSD от других основанных на BSD UNIX свободных ОС (таких как NetBSD, FreeBSD) – изначальная ориентированность проекта на создание наиболее лицензионно чистой и безопасной свободной операционной системы.

Наиболее популярным (хотя далеко не единственным) применением OpenBSD являются системы защиты сетей (межсетевые экраны). В немалой степени этому способствуют дочерние проекты, разрабатываемые параллельно.

NetBSD – это свободная, быстрая, безопасная, и очень переносимая Unix–подобная Операционная система с открытым исходным кодом. Основана на BSD UNIX®. Она доступна для широкого диапазона платформ от крупномасштабных серверов и мощных настольных систем до карманных и встроенных устройств. Её чистый дизайн и расширенные функции делают её превосходной для использования и в среде производства и исследования, а исходный код в свободном доступе в соответствии с дружественной для бизнеса лицензией. NetBSD разрабатывается и поддерживается крупным и ярким международным сообществом. Много приложений легко доступны через инструмент pkgsrc – Набор пакетов NetBSD.

22 сентября 2014 был выпущен релиз NetBSD 6.1.5, пятое безопасное и фиксирующее баги обновление NetBSD 6.1. Оно представляет выбранное подмножество исправлений, которые считаются важными по причинам устойчивости или безопасности.

#### 3.4.2.5.4 QNX

QNX — POSIX-совместимая операционная система реального времени, предназначенная преимущественно для встраиваемых систем. Считается одной из лучших реализаций концепции микроядерных операционных систем (QNX основана на идее работы основной части своих компонентов, как небольших задач, называемых сервисами, что позволяет пользователям (разработчикам) отключить любую ненужную функциональность, не изменяя ядро. Для этого можно просто не запускать определённый процесс.)

QNX сейчас способна работать практически на любом современном процессоре, используемом на рынке встраиваемых систем. Среди них присутствуют семейства x86, MIPS, PowerPC, а также специализированные семейства процессоров, такие, как SH-4, ARM, StrongARM и xScale.

Система платная, стоимость лицензии высокая, но для некоммерческого использования и для обучения она предлагается бесплатно на 30 дней. Также ограничением на широкое использование QNX является сильная зависимость от QNX Software Systems в плане лицензирования разработанного программного обеспечения. Но при этом QNX занимает лидирующую позицию среди ОС реального времени на платформе ПК.

#### 3.4.2.5.5 OS X

OS X (Mac OS X до версии 10.7 включительно) — проприетарная операционная система производства Apple. Это вторая по популярности операционная система в мире (после Windows). Её рыночная доля в сегменте настольных компьютеров в июле 2014 года составляла 7,45 %. Семейство OS-UNIX, Исходный код-Закрытый (со свободными компонентами), Последняя версия-10.10 — 16 октября 2014

OS X поддерживает следующие платформы:

- а) PowerPC: 10.0 — 10.5
- б) x86: 10.4 — 10.6
- в) x86-64: 10.4 — 10.10

В OS X используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD. До версии 10.3 OS X работала только на компьютерах с процессорами PowerPC. Выпуски 10.4 и 10.5 поддерживали как PowerPC-, так и Intel-процессоры. Начиная с 10.6 OS X работает только с процессорами Intel. OS X отличается более высокой стабильностью, чем предшественница – Mac OS 9.

В OS X (как и в любой UNIX-системе) используется вытесняющая многозадачность и защита памяти, позволяющие запускать несколько изолированных друг от друга процессов, каждый из которых не может прервать или модифицировать все остальные. На архитектуру OS X повлияла OpenStep, которая была задумана как портируемая операционная система (например, NeXTSTEP была

портирована с оригинальной платформы 68k компьютера NeXT до приобретения NeXTSTEP компанией Apple). Аналогичным образом OpenStep была портирована на PowerPC в рамках проекта Rhapsody.

Основами OS X являются:

- а) Подсистема с открытым кодом — Darwin (ядро Mach и набор утилит BSD).
- б) Среда программирования Core Foundation (Carbon API, Cocoa API и Java API).
- в) Графическая среда Aqua (QuickTime, Quartz Extreme и OpenGL).
- г) Технологии Core Image, Core Animation CoreAudio и CoreData.

Последнее время мобильная операционная система Apple iOS и настольная система OS X стали рассматриваться в компании Apple как одна система с расширяющейся функциональностью. В связи с этим существуют сведения о переносе настольной операционной системы OS X на процессоры Apple Ax с ARM-архитектурой.

В OS X используется графический интерфейс Aqua - закруглённые углы окон, полупрозрачные элементы и светлые полоски.

### **3.4.2.6       Принятое решение по типу ОС**

В качестве операционной системы предлагается использовать Linux по целому ряду причин:

- а) ОС Linux не является проприетарной системой и ее использование не накладывает дополнительной нагрузки на бюджет проекта
- б) Данная ОС имеет открытый код позволяет при необходимости собрать собственное ядро, отвечающее исключительно стоящим задачам.
- в) ОС Linux содержит в себе все необходимые сервисы для работы с современным оборудованием и сетевыми протоколами.

- г) Linux поддерживает широкий спектр аппаратных платформ: IA-32, x86-64, ARMv7, ARM64, Power

Сборке Ubuntu было отдано предпочтение в виду наличия необходимых знаний о ней среди разработчиков.

### **3.4.3 Подсистема симуляции распределенных электронных потребительских устройств (Internet of Things)**

#### **3.4.3.1 Назначение подсистемы**

Подсистема симуляции распределенных электронных потребительских устройств разработана для моделирования в режиме реального времени показаний устройств, приближенных к показаниям реальных устройств. Целью данной подсистемы является только моделирование частоты и схемы данных показаний для экспериментального исследования ЭО ПАП, но не создание моделей максимально приближенных к реальности.

#### **3.4.3.2 Типы моделируемых распределенных электронных потребительских устройств**

Для проведения экспериментальных исследований ЭО ПАП были разработаны модели следующих устройств:

- а) общедомовые счетчики тепловой энергии в жилых домах города
- б) уличные датчики температуры, устанавливаемые на улицах города

#### **3.4.3.3 Метод агентного моделирования**

В данной подсистеме для моделирования показаний электронных потребительских устройств применяется метод агентного моделирования.

Агентное моделирование (далее - АМ) это класс расчетных моделей для симуляции действий и взаимодействий автономных агентов, цель которых изучить влияние этих взаимодействий на всю систему в целом.

В агентном моделировании, система моделируется как набор автономных единиц, способных самостоятельно принимать решения, называемых агентами. Каждый агент индивидуально оценивает собственное состояние и положение и на основе оценки принимает решение, в основе которого лежит набор заданных правил. Агенты могут вести себя по-разному, в зависимости от того, какую именно систему они представляют, например они могут продавать, потреблять или производить. В основе основной идеи агентного моделирования лежат повторные взаимодействия агентов, сопернического характера. Реализация этой идеи возложена на способности современных компьютеров, что позволяет изучать динамику системы в недоступном для чисто математических методов русле.

На самом простом уровне агентная модель состоит из системы агентов и отношений/взаимодействий между ними. Даже простая агентная модель может обнаруживать сложные поведения и давать важную информацию о динамике систем реального мира, которые она моделирует. В дополнение к этому, агенты способны развиваться, проявляя непредвиденное поведение. Сложные АМ иногда включают в себя нейронные сети, эволюционные алгоритмы или другие технологии обучения, для симуляции реальных процессов обучения и адаптации.

Достоинствами агентного моделирования и его преимущества над другими техниками моделирования можно свести к трем утверждениям:

- а) АМ выявляет неожиданные (новые) свойства системы
- б) АМ показывает систему естественным образом
- в) АМ является гибкой методологией

Гибкость АМ можно рассматривать в различных перспективах. Например, добавление новых агентов в модель не составляет никакой сложности. Агентное моделирование также предоставляет естественную среду для описания сложности агентов: поведение, степень рациональности, способность учиться и развиваться и правила взаимодействия.

Другая сфера гибкости АМ заключается в способности изменять уровни описания и агрегации: можно задавать правила поведения как одному, так и группе агентов. Описывать свойства агентов на разных уровнях точности, с различной степенью подробности (опять же, как для одного так и для целой группы). Свойства агентов, правила их поведения можно улучшать, изменять, с легкостью уточнять.

#### **3.4.3.4 Программная библиотека агентного моделирования**

Подсистема симуляции использует платформу MaDKit 5 (The Multiagent Development Kit) (<http://www.madkit.net/>) для программирования агентов и их окружения.

MaDKit является платформой для разработки мультиагентных систем на языке Java. Она разработана для простого создания распределенных приложений и симуляций на основе агентного моделирования, и реализует следующую функциональность:

- а) создание агентов и управление их жизненным циклом
- б) организационная инфраструктура для коммуникации между агентами
- в) высокая гетерогенность агентов: модель агентов не предопределена жестко
- г) инструменты для создания мультиагентных симуляций
- д) инструменты для создания распределенных приложений

По сравнению с традиционными подходами, которые в основном являются агенто-центрическими, MaDKit реализует организационно-центрический подход [138], в котором модель агента не предопределена. Поэтому MaDKit создан на основе модели «агент/группа/роль»: агенты играют роли в группе и тем самым создают искусственные сообщества.

#### **3.4.3.5 Модель общедомового счетчика тепловой энергии**

Общедомовой счетчик тепловой энергии является сложной системой сенсоров, которые измеряют различные характеристики отопительной системы

дома, такие как объем теплоносителя на входе и выходе, плотность теплоносителя, температуру и т.д.

Для целей экспериментального исследования ЭО ПАП были выбраны следующие характеристики, которые моделируются данной подсистемой:

- a) суммарное количество гигакалорий потребленное домом, т.е. количество теплоты, которое было потреблено для отопления дома с начала работы счетчика
- б) температура энергоносителя поступающего в дом

Для разработки модели общедомовых счетчиков тепловой энергии были получены архивные суточные и часовые показания счетчиков за 2014-й год с нескольких жилых домов в г. Санкт-Петербург. Также на основе данных портала «Наш Петербург» (<http://gorod.gov.spb.ru>) была получена информация о количестве квартир, комнат и общей площади в данных домах. В итоге полученные данные позволяют вычислить соотношение количества квартир и комнат в доме с объемами потребления тепловой энергии, что в свою очередь позволяет моделировать показаний общедомовых счетчиков тепловой энергии различных домов.

На основе описанных выше архивных показаний были вычислены показания для отопительного сезона 2014 года (Ноябрь-Июнь), в соответствии с таблицей 3.7:

- a) среднесуточный объем потребления тепловой энергии на 1000 кв.м. общей площади здания
- б) среднесуточная температура теплоносителя

Таблица 3.7 – Среднесуточные показатели общедомового счетчика тепловой энергии

Месяц	Среднесуточное потребление (Гкал) на 1000 кв.м.	Среднесуточная температура теплоносителя (С)
Ноябрь	0,347±0,038	67,90±2,12

Продолжение таблицы 3.7

Месяц	Среднесуточное потребление (Гкал) на 1000 кв.м.	Среднесуточная температура теплоносителя (C)
Декабрь	0,453±0,104	72,31±2,54
Январь	0,521±0,184	80,18±5,01
Февраль	0,51±0,129	73,43±2,72
Март	0,503±0,12	70,58±2,81
Апрель	0,279±0,095	65,03±2,76
Май	0,216±0,106	62,60±2.1
Июнь	0,063±0,279	60,06±2,01

Жизненный цикл агента, который моделирует работу общедомового счетчика, состоит из 7 шагов, в соответствии с рисунком 3.2:

- Инициализация. На данном этапе агент получает информацию месяце и дне месяца, который ему необходимо симулировать, и получает из базы данных соответствующую информацию.
- Пауза перед стартом. Перед стартом цикла агента генерируется время паузы, которое измеряется волях задержки между показаниями, т.е. от 0 до 100% заданной задержки. Необходимость в такой паузе заключается в обеспечении асинхронности смены показаний.
- Задержка между показаниями. Необходима для контроля частоты обновления показаний.
- Генерация показаний теплопотребления. Для генерации используется нормальное распределение в диапазоне доверительного интервала для каждого из месяцев.

- д) Генерация показаний температуры теплоносителя. Аналогично предыдущему шагу.
- е) Запись показаний в БД.
- ж) Проверка на окончания жизненного цикла агента. На данном шаге проверяется получение сообщения о завершении симуляции.

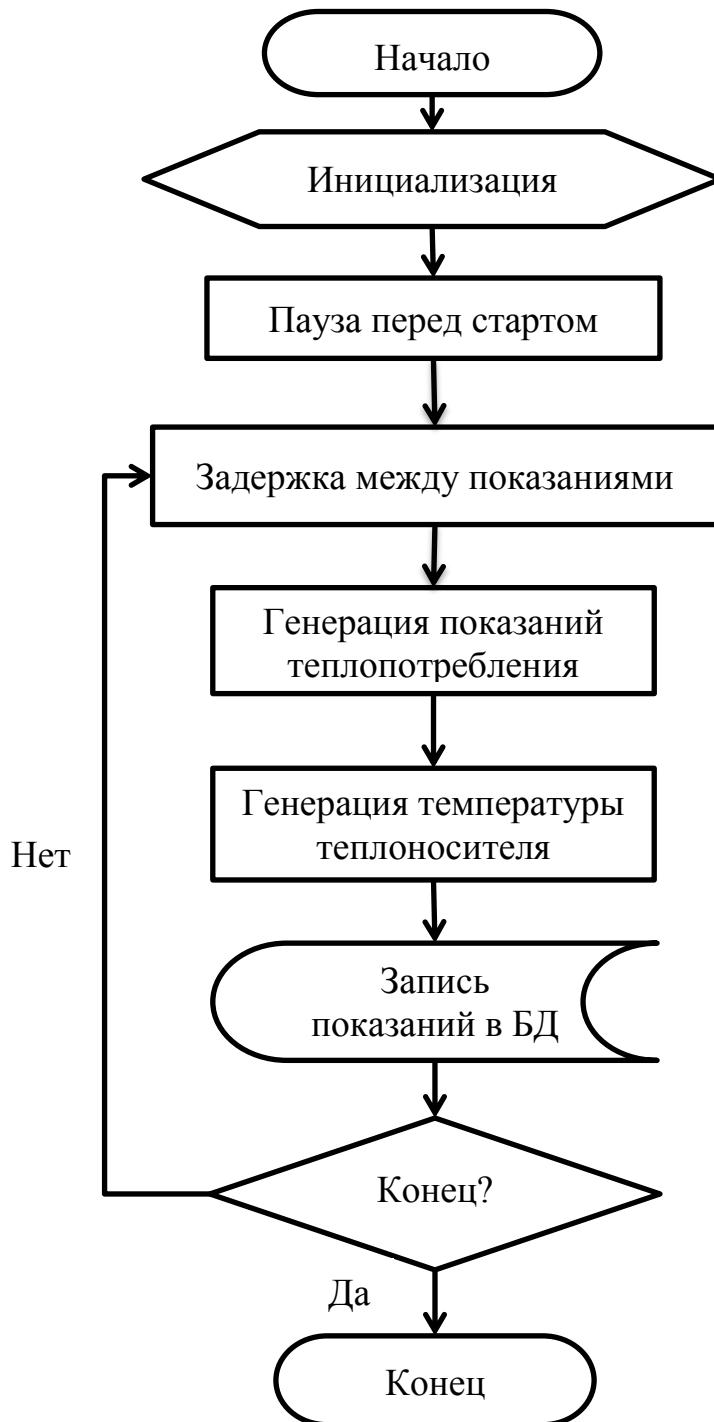


Рисунок 3.2 – Блок-схема алгоритма жизненного цикла агента

### **3.4.3.6 Модель уличного датчика температуры**

Уличные датчики температуры, как правило, позволяют измерять температуру окружающей среды в широком диапазоне температур, от -50 до +100 градусов цельсия. В целях ПАС подсистема симулирует показания температуры окружающей среды для каждого из домов, которые также симулируются данной подсистемой.

Для подготовки среднесуточных показаний температуры окружающей среды в г. Санкт-Петербург использовались архивы с портала «rp5.ru» (<http://rp5.ru>) за 2014 год. Так как показания температур для каждого из домов в реальности немного отличаются друг от друга, то для этого был принят условный доверительный интервал равный 2 градусам.

Жизненный цикл агента, который моделирует показания уличного температурного датчика, состоит из 6 шагов:

- а) Инициализация. На данном этапе агент получает информацию о месяце и дне месяца, который ему необходимо симулировать, и получает из базы данных соответствующую информацию
- б) Пауза перед стартом. Аналогично соответствующему шагу предыдущего агента
- в) Задержка между показаниями. Такая задержка необходима для контроля частоты обновления показаний.
- г) Генерация показания температуры. Для генерации используется нормальное распределение в диапазоне доверительного интервала для каждого из месяцев.
- д) Запись показания в БД.
- е) Проверка на окончания жизненного цикла агента. На данном шаге проверяется получение сообщения о завершении симуляции.

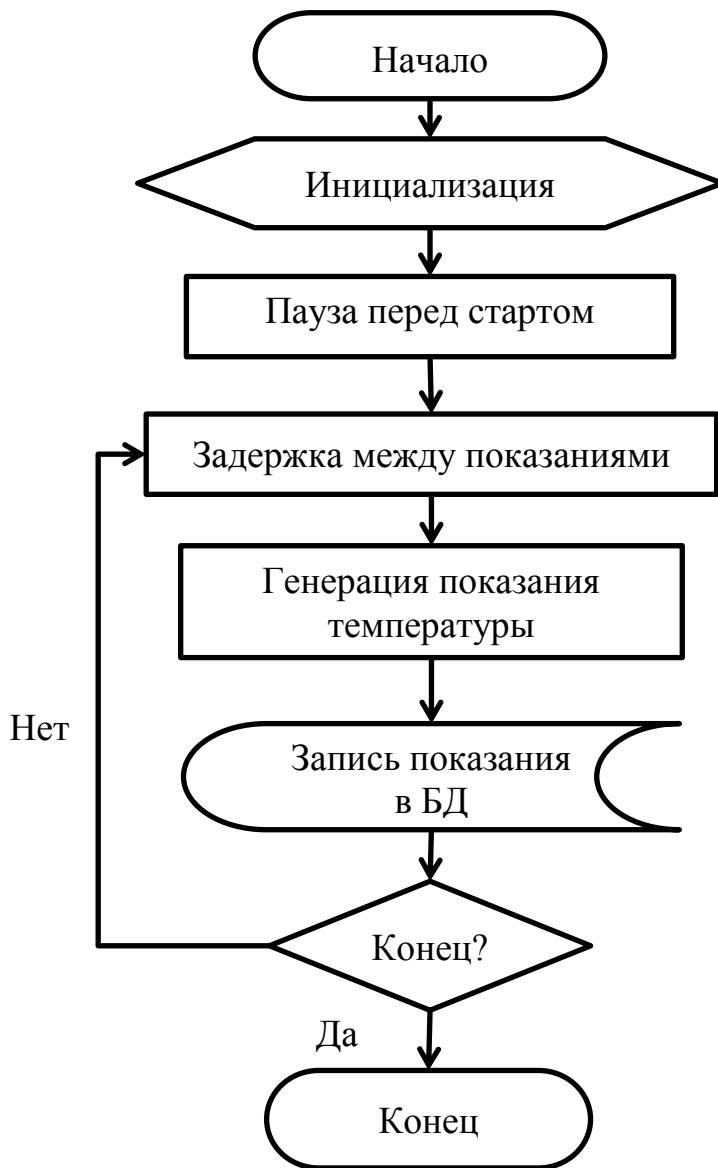


Рисунок 3.3 – Блок-схема алгоритма жизненного цикла агента

#### 3.4.3.7 Конфигурирование подсистемы симуляции

Для конфигурирования работы подсистемы симуляции распределенных электронных потребительских устройств используется файл конфигурации в формате Java properties. Для чтения данного файла конфигурации используется библиотека Owner (<http://owner.aeonbits.org>), которая по сравнению со стандартным API данная библиотека предоставляет следующие возможности:

- загрузка конфигурации из разных источников
- поддержка различных стратегий загрузки

- в) автоматическая конвертация значений конфигураций в типы данных Java
- г) поддержка расширения переменных в значения конфигураций

### **3.4.4 Подсистема хранения текущих показаний моделируемых электронных потребительских устройств**

#### **3.4.4.1 Назначение подсистемы**

В соответствии с рисунком 3.1, подсистема хранения текущих показаний моделируемых электронных потребительских устройств (далее - СУБД) является связующим звеном между подсистемой симуляции и подсистемой интеграции. Таким образом, показания сенсоров, поступающие из подсистемы симуляции сохраняются в СУБД. Затем, при появлении запросов от внешних систем подсистема интеграции формирует SQL-запрос в терминах модели данных СУБД и изымает их из СУБД.

#### **3.4.4.2 Механизмы хранения конфигурации и текущих показаний**

При построении ПАС командой выявлена необходимость записывать и хранить текущие моделируемые показания всех устройств. Для этого в ПАС предусмотрен специализированный механизм. При рассмотрении вариантов реализации данного механизма были выдвинуты два базовых направления:

- а) Создание механизма хранения на базе XML-файла.
- б) Создание механизма хранения на базе СУБД.

Основным преимуществом реализации механизма хранения на базе файлов можно считать сравнительно быстро и просто достижимый результат. Тем не менее, структура данных файла по мере реализации проекта может сильно расширяться и могут потребоваться значительные по объему описания. Хранение объемных данных в файле может быть сопряжено с некоторыми неудобствами:

- а) Передача больших по объему файлов по сети может быть затруднена в реальных условиях (симулируются именно реальные условия)

- б) Запросы к отдельным частям файлов могут выполняться довольно медленно. В некоторых случаях в разы медленнее, чем запрос к СУБД
- в) Довольно трудно разграничить права программного доступа к файлу
- г) При желании хранить дополнительные данные понадобится создание еще одного файла. При этом в СУБД имеется возможность хранить всю полноту данных в единой среде.

Следует также учесть, что предполагается хранить показания большого количества моделируемых устройств. Такой архив может занимать довольно значительное пространство на диске. Если он будет представлен в виде файла или нескольких файлов, то процесс выборки каких-то отдельных показаний будет занимать значительное время и замедлять работу всей системы в целом. В системах хранения, реализованных на основе современными СУБД существует для таких задач специализированный механизм – индексирование записей.

С учетом перечисленных выше соображений, в качестве базового механизма хранения была СУБД.

#### **3.4.4.3 Выбор конкретной реализации СУБД**

На рынке существует множество СУБД, работающих в соответствии с разными базовыми принципами: реляционные, объектные, документо-ориентированные и т.д.

В качестве реализации СУБД команда проекта остановила свой выбор именно на MySQL ввиду того, что это, во-первых, свободная, а во-вторых реляционная система управления базами данных. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Изначально, торговая марка принадлежала шведской компании MySQL AB, затем - Sun Microsystems. В настоящее время, разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку после поглощения Sun Microsystems. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

#### **3.4.4.4 Схема базы данных**

Для хранения конфигурации и текущих моделируемых показаний устройств используются 6 таблиц, в соответствии с рисунком 3.4:

- а) Сенсоры – содержит список всех устройств показания, которых моделируются
- б) Типы показаний – содержит список типов показаний, например, объём потребленного теплоносителя в гигокалориях или температура считаная с уличного датчика температуры
- в) Типы показаний сенсора – вспомогательная таблица для перечисления всех типов показаний, которые устройство генерирует
- г) Типы сенсоров – перечисление типов устройств, показания которых симулирует подсистема симуляции, например, общедомовой счетчик тепловой энергии
- д) Показания – список показаний устройств, который включает дату и время показания, значение, идентификатор сенсора и тип показания
- е) Среднесуточные значения показаний – таблица содержит средние показания и их доверительные интервалы, на основе которых генерируются моделируемые показания устройств

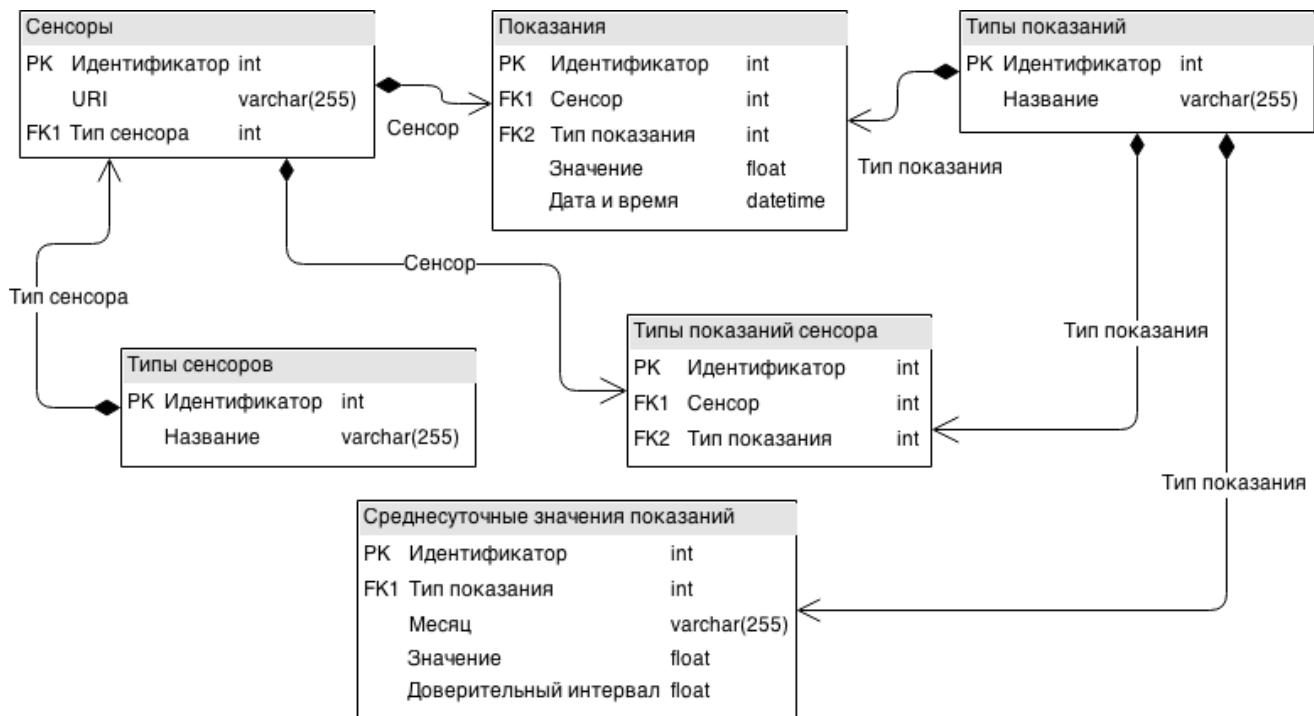


Рисунок 3.4 – Физическая схема базы данных

#### 3.4.4.5 Библиотека для взаимодействия с СУБД

Для взаимодействия с СУБД из подсистемы симуляции и подсистемы интеграции был выбрана библиотека Hibernate (<http://hibernate.org>) являющаяся наиболее популярной ORM (Object oriented mapping) библиотекой для языка программирования Java. Hibernate является реализацией стандарта JPA (Java Persistent API) и предоставляет следующие возможности:

- автоматическая генерация первичных ключей
- высокую производительность по сравнению с аналогами
- объектно-ориентированный язык запросов
- и т.д.

#### 3.4.5 Подсистема интеграции больших массивов гетерогенных данных

##### 3.4.5.1 Назначение подсистемы

Данная подсистема предназначена для предоставления унифицированного доступа к текущим показаниям симулируемых электронных потребительских устройств через программный интерфейс доступа.

### **3.4.5.2 Метод передачи данных от электронных потребительских устройств к ЦОД**

Как правило, электронные потребительские устройства в Интернете вещей имеют ограниченные вычислительные возможности, в результате к протоколам передачи данных с таких устройств предъявляются специальные требования, такие как низкие затраты на кодирование и декодирование заголовков пакетов, низкие требования к вычислительным ресурсам устройств и т.д.

Подробное описание протоколов и стеков протоколов, которые соответствуют необходимым требованиям и применяются для передачи данных в распределенной сети электронных потребительских устройств, приведено в разделе 2.5 данного отчета о ПНИ.

В целях разработки ПАС был выбран протокол прикладного уровня СоAP (Constrained Application Protocol), который работает поверх протокола UDP. Протокол СоAP имеет следующие возможности и преимущества:

- а) соответствие принципа архитектурного дизайна REST, что позволяет снизить затраты на проксирование запросов в протокол HTTP
- б) низкие затраты на кодирование и декодирование заголовков запросов
- в) поддержка URI и типов контента
- г) поддержка обнаружения ресурсов
- д) поддержка подписки на обновления состояния ресурсов и получения уведомлений в асинхронном режиме
- е) поддержка кэширования данных на основе заголовка max-age

В соответствии с таблицей 3.8, на данный момент существует несколько реализаций протокола СоAP. На основе анализа существующих реализаций и их характеристик была выбрана реализация Californium, так как она имеет следующие характеристики:

- a) реализована на языке программирования Java, что соответствует требованиям ТЗ
- б) предоставляет реализации клиента и сервера
- в) распространяется под открытой лицензией
- г) реализует протокол DTLS (Datagram Transport Layer Security), который необходим для шифрования данных

Таблица 3.8 – Реализации протокола CoAP

Название	ЯП	Клиент / Сервер	Реализованные возможности CoAP	Лицензия	Ссылка
Californium	Java	оба	Observe, Blockwise Transfers, DTLS	EPL + EDL	<a href="https://www.eclipse.org/californium">https://www.eclipse.org/californium</a>
CoAP for Go	Go	оба	Draft Subscribe	MIT	<a href="https://github.com/dustin/go-coap">https://github.com/dustin/go-coap</a>
CoAP.Net	C#	оба	Observe, Blockwise Transfers	BSD	<a href="https://github.com/smashlink/CoAP.NET">https://github.com/smashlink/CoAP.NET</a>
CoAPSharp	C#	оба	Observe, Blockwise Transfers, RD	LGPL	<a href="http://www.coapsharp.com">http://www.coapsharp.com</a>
CoAPthon	Python	клиент/сервер/прокси	Observe, Multicast discovery, CoRE Link Format	BSD	<a href="https://github.com/Tanganelli/CoAPthon">https://github.com/Tanganelli/CoAPthon</a>
Copper	Java script	клиент	Observe, Blockwise Transfers	BSD	<a href="https://github.com/mkovatsc/Copper">https://github.com/mkovatsc/Copper</a>
jCoAP	Java	оба	Observe, Blockwise Transfers	Apache License 2.0	<a href="https://code.google.com/p/jcoap/">https://code.google.com/p/jcoap/</a>
libcoap	C	оба	Observe, Blockwise Transfers	BSD/GPL	<a href="http://sourceforge.net/projects/libcoap/develop">http://sourceforge.net/projects/libcoap/develop</a>
Ruby coap	Ruby	оба	Observe, Blockwise Transfers, RD	Коммерческая	<a href="https://github.com/nning/coap">https://github.com/nning/coap</a>

### **3.4.5.3      Формат представления данных**

Данная подсистема реализует интеграцию гетерогенных данных распределенной сети электронных потребительских устройств, которая заключается в использовании RDF модели данных и использование верхнеуровневых онтологий для аннотирования данных. В разделе 2.3 данного отчета производится обзор применения данного подхода.

Для представления показаний общедомового счетчика тепловой энергии использовалась онтология SSN (Semantic Sensor Networks Ontology, <http://purl.oclc.org/NET/ssnx/ssn#>), разработанная в рамках рабочей группы W3C Semantic Sensor Network Incubator Group (<http://www.w3.org/2005/Incubator/ssn/>). Данная онтология, в соответствии с рисунком 3.5, содержит классы и свойства для представления различной информации, начиная от показаний счетчиков до места их расположения и требований к эксплуатации.

Так как онтология SSN является верхнеуровневой и покрывает только концепции, которые являются общими для всех областей применения, то для полного описания показаний общедомового счетчика тепловой энергии была разработана онтология расширяющая классы и свойства онтологии SSN, названная TEM (Thermal Energy Meters Ontology).

Онтология TEM содержит новые классы и свойства, такие как:

- а) класс tem:ThermalEnergyObservation являющийся подклассом ssn:Observation
- б) класс tem:ThermalEnergySensorOutput являющийся подклассом ssn:SensorOutput
- в) класс term:ThermalEnergyTemperature являющийся подклассом ssn:Observation
- г) класс term:ThermalCalorie являющийся подклассом ssn:UnitOfMeasure
- д) и другие

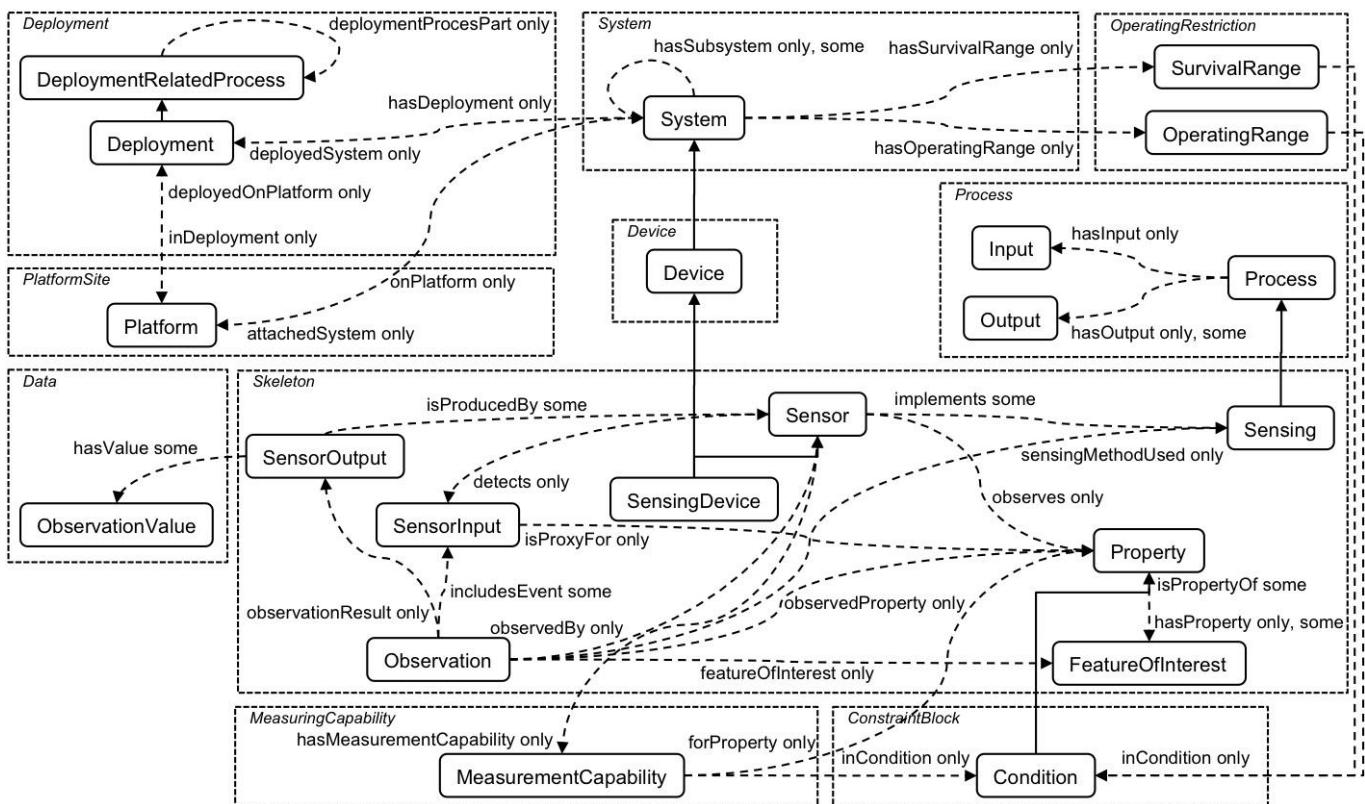


Рисунок 3.5 – Диаграмма основных классов и свойств онтологии SSN

Каждое показание устройства имеет свой URI, который выступает глобальным уникальным идентификатором. Пример такого URI:  
[http://example.com/observation#thermalenergy-\[серийный номер счетчика\]-\[время\]](http://example.com/observation#thermalenergy-[серийный номер счетчика]-[время]).

На рисунке 3.6 представлен пример представления показания объема тепловой энергии общедомового счетчика с помощью онтологий SSN и TEM.

Также аналогично было разработана онтология AIRT (Air Temperature Ontology) расширяющая онтологию SSN для представления показаний уличного датчика температуры. На рисунке 3.7 представлен соответствующий пример.

```

<http://example.com/observation#thermalenergy-123123123-11111111>
    a tem:ThermalEnergyObservation ;
    ssn:observationResultTime "11111111"^^xsd:dateTime ;
    ssn:observedBy <http://example.com/meters#123123123> ;
    ssn:observationResult      <http://example.com/observation#thermalenergy-
123123123-11111111-result> .
<http://example.com/observations#thermalenergy-123123123-11111111-result>
    a tem:ThermalEnergySensorOutput ;
    ssn:isProducedBy <http://example.com/meters#123123123> ;
    ssn:hasValue      <http://example.com/observations#thermalenergy-123123123-
11111111-resultvalue> .
<http://example.com/observations#thermalenergy-123123123-11111111-resultvalue>
    a tem:ThermalEnergyValue ;
    tem:hasQuantityValue "0.347"^^xsd:float .

```

Рисунок 3.6 – Пример показания общедомового счетчика аннотированного онтологиями SSN и TEM

```

<http://example.com/observation#airtemp-145145145-11111111>
    a airt:AirTemperatureObservation ;
    ssn:observationResultTime "11111111"^^xsd:dateTime ;
    ssn:observedBy <http://example.com/meters#123123123> ;
    ssn:observationResult <http://example.com/observation#airtempm-145145145-
11111111-result> .
<http://example.com/observations#airtempm-145145145-11111111-result>
    a airt:AirTemperatureSensorOutput ;
    ssn:isProducedBy <http://example.com/meters#145145145> ;
    ssn:hasValue      <http://example.com/observations#airtemp-145145145-
11111111-resultvalue> .
<http://example.com/observations#airtemp-145145145-11111111-resultvalue>
    a airt:AirTemperatureValue ;
    airt:hasQuantityValue "0.347"^^xsd:float .

```

Рисунок 3.7 – Пример показания уличного датчика аннотированного онтологиями SSN и AIRT

#### **3.4.5.4 Алгоритм обработки запроса на получение показаний**

Как описано в разделе 3.4.5.2, для передачи данных используется протокол CoAP и его реализация на языки программирования Java под названием Californium,

а также архитектурный стиль REST. REST-интерфейс имеет следующую архитектуру:

- a) GET /.well-known/core – возвращает информацию о существующих ресурсах и их путях
- б) GET /sensor/[номер] – возвращает информацию о сенсоре по его номеру, аннотированную с помощью онтологии SSN
- в) GET /sensor/[номер]/[название показания] – возвращает последнее показание сенсора по названию показания, например, /sensor/123123123/temperature возвращает показание по температуре энергоносителя.

Обработка запросов реализована через вызовы функций обратного вызова соответствующих пути. На рисунке 3.8 представлен общий алгоритм обработки запроса на получение показаний.

На вход алгоритм получает экземпляр класса CoAPExchange, через который можно получить значения номера сенсора и названия показания. Далее отправляет SELECT запрос в подсистему хранения для того чтобы получить по номеру сенсора и названию показания последнее показание. Если такого показания не существует или не существует сенсора с данным номером, то в ответ возвращается код 4.04, который эквивалентен к HTTP коду 404. Если необходимые данные найдены, то на основе статического шаблона RDF данных создается набор RDF данных через функцию замены подстроки в строке и полученная информация отправляет в ответу клиенту.

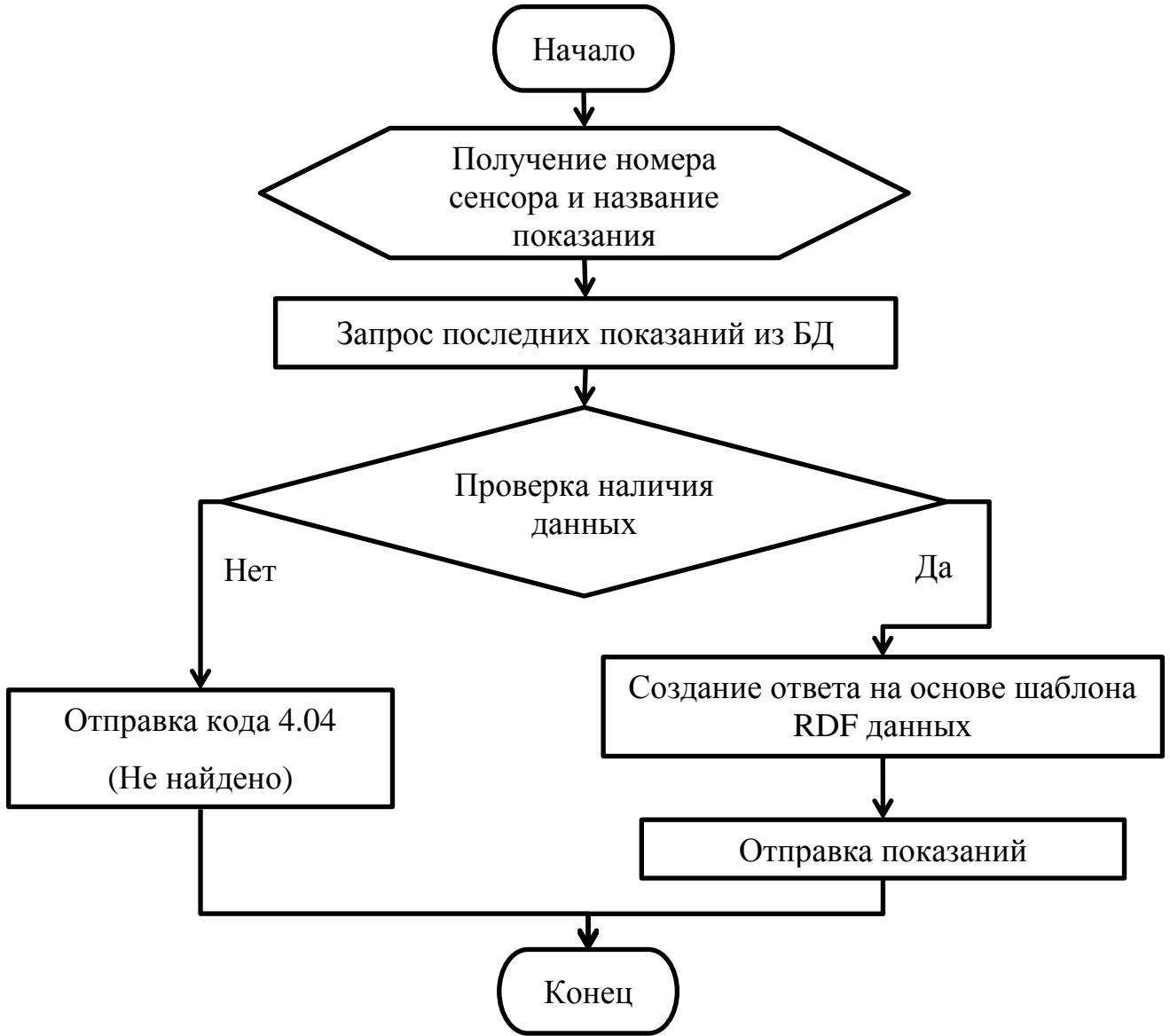


Рисунок 3.8 – Пример обработки запроса на получение последних показаний

### 3.5 Результаты

В данном разделе отчета описан разработанный программно-аппаратный стенд с подсистемой виртуальных серверов, подсистемой интеграции больших массивов гетерогенных данных и симуляции распределенных электронных потребительских устройств для проведения экспериментальных исследований экспериментального образца масштабируемой сервис-ориентированной программно-аппаратной платформы для сбора, нормализации, обработки и визуализации больших массивов гетерогенных данных в распределенной сети электронных потребительских устройств.

В соответствие с ТЗ, ПАС должен обладать достаточными ресурсами для симуляции 1000 и более сенсоров. Программные и аппаратные решения, отобранные для реализации ПАС, обеспечивают солидный запас по мощности, пропускной способности и объему хранения и, как планируется, смогут обеспечить симуляцию работы порядка 2000 сенсоров.

Также разработана техническая документация на ПАС и представлена в Приложениях:

- а) Приложение Д. Руководство по эксплуатации ПАС
- б) Приложение Е. Структурная схема ПАС
- в) Приложение Ж. Формуляр
- г) Приложение И. Описание программного обеспечения ПАС
- д) Приложение К. Текст программного обеспечения ПАС

В результате работы был создан программно-аппаратный стенд, соответствующий требованиям технического задания.

# **4    Анализ существующих приемов онтологического инжиниринга и технологий семантического веба для агрегации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных**

## **4.1    Обзор существующих трендов**

Несколько лет назад комбинация распределенной сети электронных потребительских устройств с технологиями семантического веба, веб сервисами, технологиями сети Интернет и технологиями баз данных назвали сенсорной сетью (Sensor Web) или сенсорным интернетом (Sensor Internet) [138, 139, 140, 141, 142]. Большая часть работ выполнены в данном направлении выполнена в рамках рабочей группы OGS Sensor Web Enablement, задача которой является создание спецификаций различной функциональности связанной с управлением сенсорными данными (наблюдения, измерения, описание сети сенсоров, датчики, потоковые данные и т.д.), и различных типов сервисов, которые обрабатывают эти источники данных (планирование, уведомления, сбор и управление наблюдениями и измерениями, и т.д.).

Также существуют работы по разработке платформ, которые предоставляют сервисы необходимые для разработки приложений использующих сенсорные данные. Такие платформы включают в себя программные библиотеки для выполнения предметно независимых управленческих задач, агрегация, выполнения запросов и т.д., и они также предоставляют предметно зависимые сервисы, например, мониторинг прибрежной зоны [110], мониторинг за состоянием пациентов [142].

И наконец, работы (например, Pachube, SensorMap) по разработке централизованных реестров сенсорных данных, задача которых регистрировать

источники данных и предоставлять доступ к ним с помощью разных интерфейсов, таких как REST-интерфейсы, веб сервисы или специализированные языки запросов.

## **4.2 Типовая архитектура программного обеспечения на основе данных электронных потребительских устройств**

На рисунке 4.1 представлена типовая архитектура приложения использующего сенсорные данные, которая может быть характеризована следующим образом:

- а) изменчивость и гетерогенность данных, устройств и сетей (в том числе ненадежные узлы и связи сети, шум, неопределенность и т.д.).
- б) использование сложных источников данных (сенсоры, изображения, ГИС-данные и т.д.) в различных ситуациях (в режиме реального времени, потоки данных, исторические данные и обработанные данные).
- в) наличие нескольких существующих административных зон.
- г) необходимость в управлении несколькими, конкурентными и не скординированными запросами к сенсорам.

Далее описываются текущие исследовательские проблемы в данной области, для которых позже будут описаны решения, основанные на применении онтологического инжиниринга и технологий семантического веба.

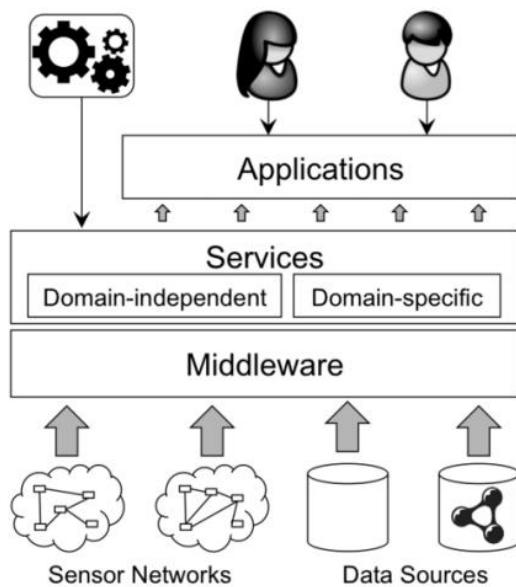


Рисунок 4.1 - Типовая архитектура приложения использующего сенсорные данные

### 4.3 Существующие исследовательские проблемы

Одной из существующих исследовательских проблем является проблема, связанная с построение некоторого абстрактного уровня, в котором собираются и анализируются сенсорные данные в целом. Сенсорные данные могут быть обработаны на низком уровне, то есть на уровне устройств и сети, в целом с использованием низкоуровневых языков программирования и операционных систем. Но они также могут обрабатываться на более высоком уровне (например, с помощью декларативных непрерывных запросов к потокам данных), тем самым скрывая от пользователей инфраструктурную и синтаксическую гетерогенность автоматические развертываемых распределенных сетей электронных потребительских устройств.

Другой проблемой является адекватная характеристика и управление качеством сенсорных данных. Проблемы связанные с недоступность некоторого «куска» данных в течении некоторого времени может иметь различные значения, когда они рассматриваются с точки зрения приложения: сенсор был недоступен, не было соответствующего события, которое бы инициировало сбор данных в данный момент времени, связь с устройством была потеряна и т.д. Другие проблемы,

например, частота получения данных может, зависят от некоторых внутренних и внешних состояний сети сенсоров. Таким образом, существуют несколько характеристик, которые связаны с качеством и которые влияют на процесс сбора данных.

Также проблемой является идентификация и обнаружение релевантных источников сенсорных данных, с помощью чего могут быть выполнены задачи интеграции и слияние данных. Количество сенсорных сетей развернутых в реальном мире постоянно растет, что обуславливается снижением цен на аппаратное обеспечение. В результате, больше экспериментов и инициатив развертывают сенсорные сети в различных областях, и нахождение правильной и релевантной информации является важным условием выполнения задач интеграции данных.

И наконец, еще одной проблемой является необходимость в разработке инструментов, повышающих скорость разработки приложений основанных на сенсорных данных, которые способные обрабатывать сенсорные данные в соответствие со всеми перечисленными выше характеристиками и проблемами. Данная проблема включает в себя задачи валидации и интеграции данных, а так же предоставление общего интерфейса и форматов данных между приложениями, базами данных и сенсорными сетями и т.д. Данная проблема требует возможности разработки приложений, работающих с разными схемами данных и качеством сервиса, и использования сенсорных данных для целей разработчика и пользователя.

#### **4.4 Применение методов онтологического инжиниринга и технологий семантического веба**

В этом разделе описываются решения перечисленных выше исследовательских проблем в существующих инициативах и проектах, с использованием технологий семантического веба и онтологического инжиниринга.

Как определено в предыдущем разделе, одной из исследовательских проблем является реализация некоего абстрактного уровня, в рамках которого происходит

сбор, обработка и дальнейшее управление сенсорными данными. Некоторое количество онтологий для сенсоров и их данных были разработаны и описаны в существующей литературе [69], цель которых описать различные аспекты сенсорных данных, как со стороны самих электронных потребительских устройств (сенсоров), так и со стороны предметной области, в которой применяются сенсоры. Некоторые аспекты, которые были описаны в этих онтологиях, являются релевантными данной проблеме. Такие как разделение на данные полученные на прямую от сенсоров («сырые» данные) и вновь полученные данные, представление такие концепций, как частота, и рассмотрение наблюдений и измерений в соответствии с моделями OGC. Представление последних концепций были описаны в работах [68, 114]. Разработка таких онтологий в данной области является одной из главных задач рабочей группы W3C Incubator Group on Semantic Sensor Networks, поэтому этой группой была разработана онтология SSN.

В упомянутой выше работе над онтологией для сенсоров и их измерений, так же уделяется внимание описанию качества источников сенсорных данных, но это не является центральной проблемой рассматриваемой в данной работе. Качество данных является большой исследовательской область, которая применима не только к сенсорным данным, но к любым данным, с которыми работают приложения. Принято обсуждать качество данных в рамках реляционных баз данных, полуструктурированных данных, данных генерируемых пользователями и т.д. Поэтому это является свойством данных в целом, а не только сенсорных данных в частности. Однако сенсорные данные в большей степени зависят от контекста сенсорной сети, такого как физическое устройство сети, стратегия развертывания, условия окружающей среды. Такой контекст влияет на качество данных (например, частота измерений) и поэтому должен учитываться, что корректно интерпретировать данные. Проводиться некоторая начальная работа по определению моделей качества данных для таких типов данных, через категоризацию существующих подходов к другим типам данных, отбор и адаптированные таких подходов в контексте сенсорных сетей. Тоже самое

относится к качеству сервиса сенсорных сетей, в терминах параметров, которые также применимы к другим типам данных, например, надежность и т.д.

Что касается интеграции и синтеза данных, то некоторая работа была проведена в контексте интеграции и синтеза гетерогенных потоковых данных. Какие-то из этих работ используют технологии семантического веба, а какие то нет. Текущим исследовательским трендом является генерация связанных данных из потоковых данных сенсорных сетей [115, 116], трансформируя сенсорные данные в RDF и открывая к ним доступ по протоколу HTTP с помощью специальных URI адресов. Это предоставит возможность реализовать «бесшовную» навигацию по сенсорным данным и данным других типов. Другие работы нацелены на реализацию механизма семантических запросов в потоковым сенсорным данным. Они используют декларативную инфраструктуру запросов к сенсорным потоковым данным аннотированным онтологиями. Такие подходы расширяют существующие языки запросов, такие как SPARQL, конструкциями, которые обычно применяются к потоковым данным (например, окна, привязанные ко времени или количеству кортежей). Примерами таких работ являются языки C-SPARQL [65] или Streaming SPARQL [117], и работы направленные на трансформацию сенсорными данными и такими языками запросов [118].

В контексте идентификации и определения местоположения релевантных сенсорных данных в реальном мире, существуют работы по определению интерфейсов регистрации сенсорных данных и разработки соответствующей программной архитектуры, которая может справиться с типами запросов, которые обычно относятся к сенсорным данным. Такие регистры должны предоставлять соответствующую поддержку пространственно-временных запросов (например, «получить источники данных, которые содержат информацию о температуре в данном регионе за последние два дня») и запросов к метаданным, связанных с существующими онтологиями сенсорных данных. Некоторые работы в данном контексте могут быть найдены в этой статье [148].

И наконец, другая проблема идентификации связана с разработкой высокоуровневых программных интерфейсов (API), которые ускоряют разработку соответствующего программного обеспечения, которое использует данные сенсорных сетей и существующих баз данных. Такие программные интерфейсы должны справляться с гетерогенными данными различных типов (статически и потоковые), поддерживать определенные выше семантические подходы, и помогать пользователям взаимодействовать и понимать потенциально ошибочные и гетерогенные данные, поступающие с сенсорных сетей. Примера таких интерфейсов является SemanticMap [120] и некоторые другие работы связанные с принятием решений в области мониторинга окружающей среды.

# **5 Анализ применимости технологии связанных данных (Linked Data) для задачи интерактивной визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурированных данных**

## **5.1 Вызовы современности, связанные с возрастающим объемом данных**

### **5.1.1 Источники возрастающих объемов данных**

В современном мире данные становятся неотъемлемой частью существования человека. Данные порождаются практически каждым действием индивида в общественном пространстве: создания документа, выход в интернет, телефонный звонок, приход к врачу и проезд в общественном транспорте. При этом с ростом количества информационных систем деятельность людей отражается и преломляется во все расширяющимся информационном пространстве и любое событие или действие становится катализатором трудно прослеживаемого цикла обновлений данных в информационных системах, которые расходятся, подобно кругам на воде от броска камня.

Ученые уже некоторое время назад начали выражать обеспокоенность в связи с непрекращающимся, а даже ускоряющимся ростом совокупного мирового объема данных. Исследования мировых лидеров в прогнозных исследованиях показывают, что до 2020 года количество информации и потребности в ней будут расти экспоненциально. Без умения создавать и обрабатывать такие объемы информации лица, принимающие решения будущего будут введены в состояние, которое можно назвать «аналитический паралич» [150]. И действительно, предсказания ученых сбываются: по результатам последнего исследования компании IDC, которая регулярно проводит исследования в сфере накопления данных, показало, что на сегодняшний день проанализировано менее 1% всей имеющейся информации,

защищено менее 20%. При этом по прогнозам IDC к 2020 г. Цифровая Вселенная достигнет объема в 40 зеттабайт, что превосходит предыдущий прогноз на 5 зеттабайт. Всего с начала 2010 года объем данных вырос в 50 раз. Очевидно, что рост мирового объема данных носит экспоненциальный характер, и вероятность замедления темпов роста приближается к нулю.

В процессе роста мирового объема данных отражены условные эпохи развития информационных технологий. До 1986 года главенствовали мейнфреймы, которые занимали огромные площади и по стоимости могли сравняться с самолетами. Число их было сравнительно малым, поэтому количество пользователей в эту эпоху исчислялось миллионами – см. рисунок 5.1. Пользователями были сверх-узкая прослойка людей, которые обладали очень специальным набором знаний для работы с ними. Интерфейсы взаимодействия были крайне не просты в использовании и ограждали более широкую аудиторию от этих технологий.

С наступлением эры персональных компьютеров произошла коренная революция. Теперь каждый человек мог пользоваться своим личным компьютером и дома, и на работе. Такой прорыв был совершен благодаря принципиально новой архитектуре аппаратной платформы в разы снизилась стоимость производства и владения компьютером. Как следствие, произошел бурный рост числа пользователей и расширение аудитории. Для обслуживания этой аудитории было выпущены тысячи и тысячи приложений, удовлетворяющие новым потребностям аудитории.

Следующим толчком, приведшим ко второй революции, можно считать появление и повсеместное распространение мобильных телефонов и их потомков – смартфонов. Теперь микрокомпьютер в кармане каждого человека обусловил зарождение новой волны создания приложений. Более того мобильные телефоны, планшетные компьютеры и носимые устройства сами дополнительными источниками данных.

По мере расширения цифровой вселенной появляются все новые устройства, которые порождают данные без участия человека – таковыми являются всевозможные системы мониторинга, системы безопасности и видеонаблюдения, системы управления производством и т.д. Более того, самые привычные для человеческого быта устройства стали превращаться в источники данных, имеющих выход в интернет – холодильники и микроволновые печи, пылесосы и даже светильники. Технологии умного дома, умного производства и умного города напрямую связаны с генерацией новых потоков данных. В целом, нужно отметить один существенный факт – темпы роста объема данных, порождаемых машинами, значительно превышают темпы роста данных, порождаемых человеком. Согласно всему исследованию, данные, порождаемые машинами станут основным фактором роста объемов данных и объем таких данных к 2020 году возрастет в 15 раз.

В 2005 году совокупная доля данных, порожденных машинами, составляла 11% и к 2020 году эта доля превысит 40%. Предполагается, что предыдущие эпохи компьютеризировали людей и предприятия, теперь же компьютеризация затронет мир вещей и общее число таких вещей на сегодняшний день достигает астрономических размеров - 200 миллиардов.

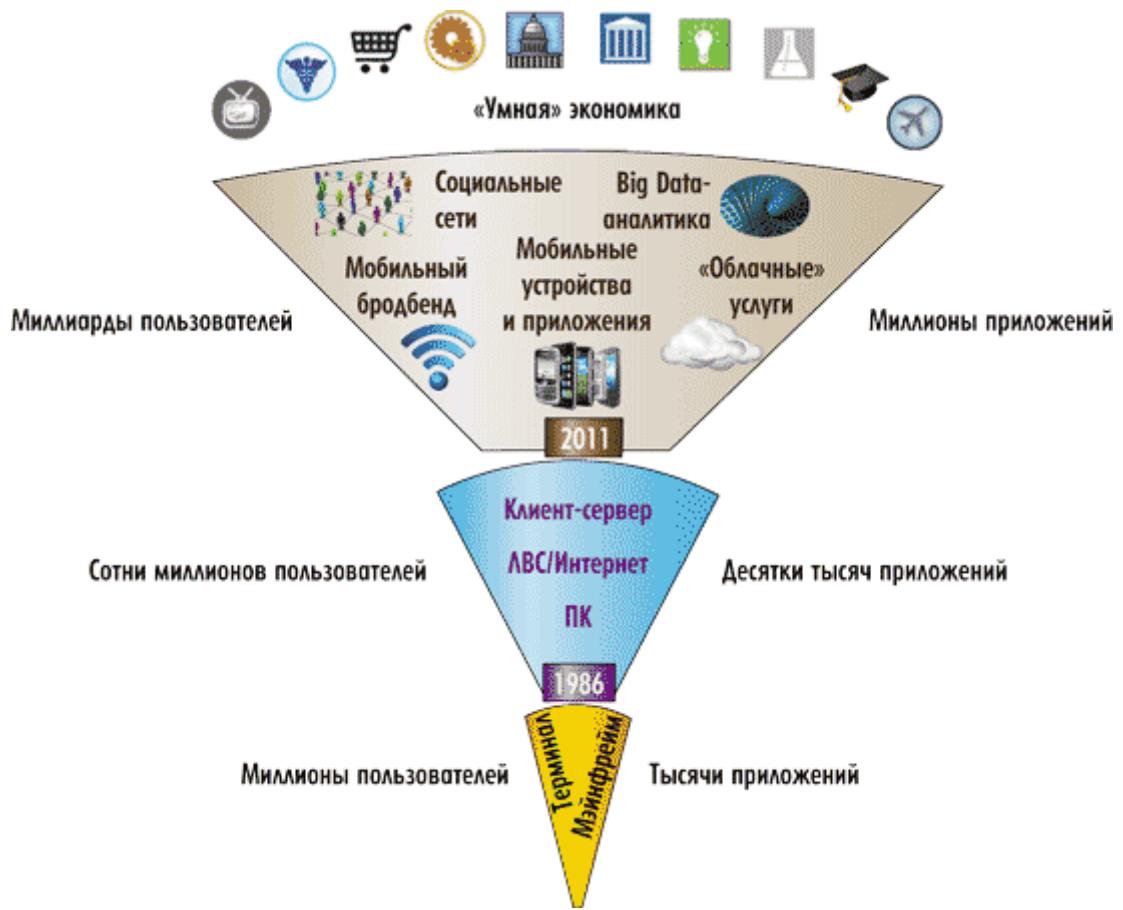


Рисунок 5.1 - Пирамида соотношения количества пользователей с количеством приложений

### 5.1.2 Данные, порождаемые человеком

В процессе роста мирового объема данных отражены условные эпохи развития информационных технологий. До 1986 года главенствовали мейнфреймы, которые занимали огромные площади и по стоимости могли сравняться с самолетами. Число их было сравнительно малым, поэтому количество пользователей в эту эпоху исчислялось миллионами – см. рисунок 5.1. Пользователями были сверх-узкая прослойка людей, которые обладали очень специальным набором знаний для работы с ними. Интерфейсы взаимодействия были крайне не просты в использовании и ограждали более широкую аудиторию от этих технологий.

С наступлением эры персональных компьютеров произошла коренная революция. Теперь каждый человек мог пользоваться своим личным компьютером и дома, и на работе. Такой прорыв был совершен благодаря принципиально новой

архитектуре аппаратной платформы в разы снизилась стоимость производства и владения компьютером. Как следствие, произошел бурный рост числа пользователей и расширение аудитории. Для обслуживания этой аудитории было выпущены тысячи и тысячи приложений, удовлетворяющие новым потребностям аудитории.

### **5.1.3 Данные, порождаемые машинами**

Следующим толчком, приведшим ко второй революции, можно считать появление и повсеместное распространение мобильных телефонов и их потомков – смартфонов. Теперь микрокомпьютер в кармане каждого человека обусловил зарождение новой волны создания приложений. Более того мобильные телефоны, планшетные компьютеры и носимые устройства сами дополнительными источниками данных.

По мере расширения цифровой вселенной появляются все новые устройства, которые порождают данные без участия человека – таковыми являются всевозможные системы мониторинга, системы безопасности и видеонаблюдения, системы управления производством и т.д. Более того, самые привычные для человеческого быта устройства стали превращаться в источники данных, имеющих выход в интернет – холодильники и микроволновые печи, пылесосы и даже светильники. Технологии умного дома, умного производства и умного города напрямую связаны с генерацией новых потоков данных. В целом, нужно отметить один существенный факт – темпы роста объема данных, порожденных машинами, значительно превышают темпы роста данных, порожденных человеком. Согласно всему исследованию, данные, порождаемые машинами станут основным фактором роста объемов данных и объем таких данных к 2020 году возрастет в 15 раз.

В 2005 году совокупная доля данных, порожденных машинами, составляла 11% и к 2020 году эта доля превысит 40%. Предполагается, что предыдущие эпохи компьютеризировали людей и предприятия, теперь же компьютеризация затронет

мир вещей и общее число таких вещей на сегодняшний день достигает астрономических размеров - 200 миллиардов.

Традиционно для описания проблемы больших данных стали применять термин Big Data, который появились в науке. В частности, сам термин ввел в оборот в 2008 году редактор журнала Nature Клиффорд Линч (Clifford Lynch) в материалах спец выпуска, целиком посвященного влиянию больших объемов данных на будущее науки и техники.

Интересно, что за 8 лет до появления термина больших данных Doug Laney в 2000 году выделил три особенности больших данных, которые окрестили 3V, а именно: Volume, Variaty, Velocity (объем, скорость поступления и вариативность). Имеется в виду то, что данные, относящиеся к Big Data, имеют значительные объем, поступают с большой скоростью и разнообразны по своей сути.

Большой объем данных – это лишь верхушка айсберга. Закономерно, что для большого объема данных нужны все возрастающие объемы хранилищ и новые технологии хранения. По большому счету здесь индустрия идет в темах с приростом объема данных: появились кластерные хранилища, широко известны твердотельные носители информации, облачные технологии хранения тоже отодвигают наступление коллапса.

Со второй характеристикой – скоростью обработки возникают значительно более серьезные трудности. Действительно, данные поступают в большом объеме в малые промежутки времени. Приведем, к примеру, всемирно известный Большой Адронный Коллайдер, сенсора и датчики которого порождают петабайты данных в секунду. В данном случае, современные системы хранения просто не в состоянии записать такой объем данных в столько короткий период времени. Вторым примером может служить любая густо населенная социальная сеть. В Facebook миллионы пользователей каждую секунду по всему миру совершают какие-либо действия: публикуют фотографии, формируют новые связи с другими аккаунтами,

комментируют публикации и т.д. Эти действия порождают терабайты данных в секунду и адекватное хранение таких данных – серьезная техническая задача.

Третья особенность Big Data – это многообразие. Действительно, растет количество сенсоров, которые порождают все новые данные. Переход к цифровой фотографии и цифровому видео добавил новые виды данных к уже имеющимся. Данные разнородны и приходят в системы хранения в неструктурированном виде. Вопрос преобразования, нормализации данных, а потом и их обработки стоит крайне остро. Большинство традиционных алгоритмов обработки данных не способно обрабатывать данные в таких объемах с необходимой скоростью.

Компания IBM, являясь мировым лидером в области обработки данных, сформулировала для проблемы Big Data в графическом виде следующим образом, в соответствии с Рисунок 5.2.

18

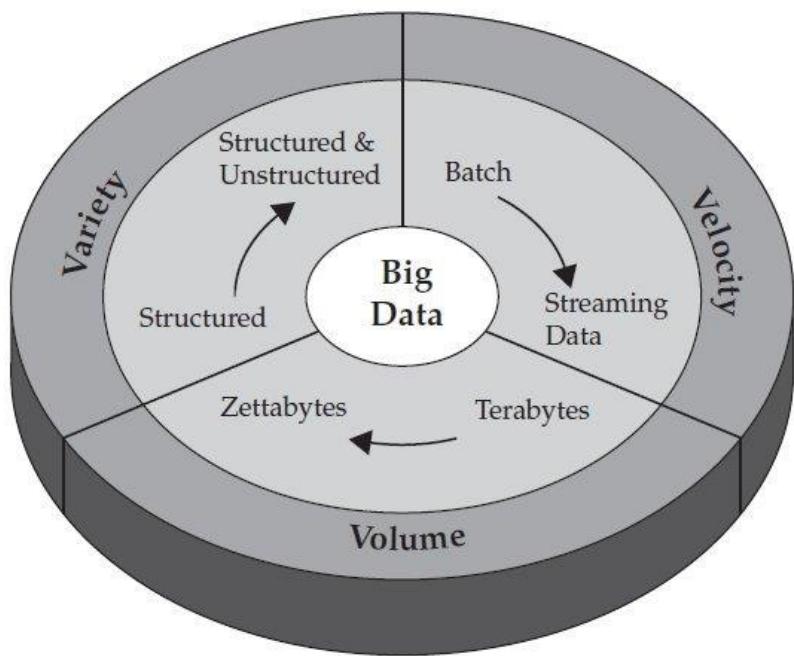


Рисунок 5.2 - Три составляющие Big Data

Видение IBM отражает тенденция в развитии проблемы Big data: возрастают объемы, данные теряют структурированность и из статичных превращаются в поточные.

В результате, по мнению исследователей из компании EMC<sup>2</sup> [151] и компании IDC на сегодняшний день анализируется 1% из всех накопленных человечеством данных, а защищено должным образом только 20% всех данных. Эти цифры поражают и самым наглядным образом иллюстрируют актуальность и глубину стоящей перед человечеством проблемы.

#### **5.1.4 Обобщение проблематики**

Проводимая ПНИ напрямую связана с вопросами обработки больших данных, потому что данные будут поступать от сенсоров в большом количестве с широкой вариативностью, и будут требовать обработки и анализа, выполняемых в реальном времени. Следовательно, все перечисленные проблемы будут иметь отношение и к заявленной ПНИ, а именно:

- а) Неуклонный рост объема данных со временем;
- б) Слабая структурированность и широкая вариативность получаемых данных;
- в) Сложность проведения обработки и анализа данных в реальном времени;
- г) Трудности с обеспечением безопасности передачи данных и ограничения доступа к ним.

### **5.2 Способы решения проблемы увеличивающихся объемов данных**

#### **5.2.1 Конвенциональные СУБД**

Реляционные или обычные, или, конвенциональные СУБД до недавнего времени были единственным возможным технологическим подходом к задаче хранения данных и обеспечения программных интерфейсов доступа к ним. По мере роста данных организации стали прибегать к следующим очевидным способам решения проблемы в пределах существующих СУБД [152]:

- a) Купить более мощное аппаратное обеспечение. Однако с стоимость оборудование возрастает для решения сложных задач не в линейной зависимости, а ближе к экспоненциальной. Объединение нескольких серверов в кластеры уже само по себе сильно затратное мероприятие, поэтому многим компаниям такое решение показалось не самым оптимальным. Покупка дорогостоящего оборудования сопровождается и ростом сопутствующих затрат, а именно: наращивание мощностей системы охлаждения, наращивание стоечного пространства, наращивание серверных площадей, дополнительное рабочее время системного администратора и т.д.
- б) Оптимизация запросов к СУБД, создание дополнительных индексов, расширения кэша на чтение и т.д. Это более сложный способ, потому что требует изменения программного кода приложений, пересмотра принципов работы клиентского ПО и предсказаний относительно развития запросов в будущем. К сожалению, рост объема данных и изменение модели хранения данных происходит быстрее, чем могут быть внедрены эти меры.
- в) Построение очередей запросов лишь выставляет запросы в очередь и не решает проблемы получения этих данных. Длина очереди будет расти, а запросы остаются либо не отвеченными своевременно, либо не отвеченными вовсе.
- г) Изменение модели данных – это крайняя мера, потому что потребует перенесения данных, изменения программного кода приложений в местах обращения к базам данных и сильно приближается к написанию приложений с нуля.

Если обобщить, то реляционные СУБД могут быть использованы для хранения больших данных только в том случае, когда известно, что рост объема данных в скором будущем остановится, а характер поступающих данных останется по большей части неизменным. На практике же никто не может дать гарантий по

перечисленным в предыдущем предложении пунктам, поэтому приходится декларировать сравнительную непригодность обычных СУБД для решения проблем Big Data.

Достоинства:

- а) Относительная прозрачность модели хранения данных
- б) Наличие огромного арсенала мощных средств по работе с реляционными СУБД (фреймворков, библиотек)
- в) Широта функционала реляционных СУБД (контроль доступа, контроль изменений, визуализация схемы данных, индексы, язык запросов SQL и т.д.)

Недостатки:

- а) Неспособность предложить адекватное решение для проблемы big data
- б) Относительно малая производительность

Если обобщить, то реляционные СУБД могут быть использованы для хранения больших данных только в том случае, когда известно, что рост объема данных в скором будущем остановится, а характер поступающих данных останется по большей части неизменным. На практике же никто не может дать гарантий по перечисленным в предыдущем предложении пунктам, поэтому приходится декларировать сравнительную непригодность обычных СУБД для решения проблем Big Data.

### **5.2.2 Специализированные СУБД**

Энтузиасты движения NoSQL выдвигают тезис о том, что реляционные базы данных как универсальное средство не подходят для решения некоторых задач, связанных с проблемой big data. Они не предлагают огульно отказаться от конвенционных СУБД, а лишь хотят развивать специализированные решения, построенные на иной архитектуре. Итак, каковы же требования к этой архитектуре?

- а) Упразднение того, что не нужно (для простых решений не нужно всего функционала реляционных СУБД)
- б) Обеспечение высокой производительности
- в) Отказ от некоторых механизмов соблюдения целостности данных
- г) Возможности неограниченного масштабирования

Сама же архитектура характеризуется следующими особенностями [124]:

- а) Уход от табличной организации данных. Данные организованы в виде пар «ключ-значение», а для поиска нужно ключа используются хеш-таблицы.
- б) Уход от традиционного языка запросов SQL, но обращение происходит только к ключам.

Ниже приведена классификация NoSQL решений с точки зрения моделей данных:

- а) Базы данных ключей-значений. Только ключи и значение и ничего более.
- б) База данных, хранящие данные в колонках. Это значит, что ключ связан с набором колонок с данными. Колонки могут иметь наименования. Этот подход наиболее близок к реляционному
- в) Документальные базы данных объединяющая несколько пар ключ-значение в единый «документ». Документы могут группироваться в иерархии.
- г) Базы данных графов предназначены для тех данных, которые по своей специфике органично группируются в графы с зависимостями. Данные представлены в виде вершин и связей между ними, которые носят ту или иную семантику.

Достоинства:

- а) Возможности увеличения производительности

- б) Неограниченные возможности масштабируемости
- в) Надежность при отказе оборудования или сетевой недоступности

Недостатки:

- а) Непрозрачность модели данных и бизнес-логики решений на NoSQL
- б) Неразвитость сопутствующих технологий

Несмотря на такие крайне важные для решения проблемы big data особенности как ускорение работы, простота и дешевизна масштабирования, возможность работать на кластерах и проч., NoSQL обладают серьезными недостатками, сдерживающими их повсеместное внедрение, в частности: непрозрачностью модели данных и затрудненной передачей знаний о предметной области через модель данных.

### **5.2.3 Big Data**

По сути, MapReduce – это новый подход к процессу обработки запросов к данным и операциям над ними. Изначально – это был алгоритм, созданный компанией Google. Операция Map разбивает все необходимые для запроса данные на блоки, за которые ответственные те или иные узлы кластера. Компоненты кластера выполняют необходимую часть операций, и результат их работы попадает, является входом в операцию Reduce, где производится вычисление общего результата.

Одной из самых известных и до сих применяемых реализаций алгоритма является Hadoop, являющимся открытым проектом.

Достоинства:

- а) Неограниченные возможности масштабирования
- б) Возможность быстрого выполнения операций, которые для реляционных БД являются затратными по времени

Недостатки:

- а) Слабая поддержка схем данных
- б) Очень узкий спектр применения
- в) Сильные ограничения, которые накладывает фреймворк на программирование его работы
- г) Трудности при множественном обращении к одному и тому же набору данных – например, в операциях машинного обучения [125].

Выгоды от использования технологий на основе MapReduce сильно нивелируются значительной ограниченностью сферы его использования.

#### **5.2.4 Linked Data**

Linked data или связанные данные – этот термин был введен сэром Тимоти Джоном Бернерс-Ли, директором консорциума W3C в рамках дискуссии о проекте Semantic Web. В статье Design Issues: Linked Data [102], он выводит 4 основных принципа связанных данных:

- а) Использование URI (Uniform Resource Identifier – унифицированный идентификатор ресурса) для определения сущностей
- б) Использование HTTP URI таким образом, чтобы на эти сущности можно было ссылаться и чтобы они могли быть найдены человеком и программным клиентом.
- в) Предоставлять полезную информацию о сущности при условии, что её URI разыменовании, используя такие стандарты, как RDF и SPARQL.
- г) При опубликовании данных в Веб включать в описание ссылки на другие сущности (при наличие взаимосвязей) используя URI этих сущностей.

Для понимания этих принципов необходимо объяснить значение некоторых специфических терминов

URI - это символьная строка, позволяющая идентифицировать какой-либо ресурс: документ, изображение, файл, службу, ящик электронной почты и т. д. Прежде всего, речь идёт, конечно, о ресурсах сети Интернет и Всемирной паутины. URI предоставляет простой и расширяемый способ идентификации ресурсов. Расширяемость URI означает, что уже существуют несколько схем идентификации внутри URI, и ещё больше будет создано в будущем.

RDF – это разработанная консорциумом Всемирной паутины модель для представления данных, в особенности — метаданных. RDF представляет утверждения о ресурсах в виде, пригодном для машинной обработки. RDF является частью концепции семантической паутины.

Ресурсом в RDF может быть любая сущность — как информационная (например, веб-сайт или изображение), так и неинформационная (например, человек, город или некое абстрактное понятие). Утверждение, высказываемое о ресурсе, имеет вид «субъект — предикат — объект» и называется триплетом. Утверждение «небо голубого цвета» в RDF-терминологии можно представить следующим образом: субъект — «небо», предикат — «имеет цвет», объект — «голубой». Для обозначения субъектов, отношений и объектов в RDF используются URI.

Множество RDF-утверждений образует ориентированный граф, в котором вершинами являются субъекты и объекты, а рёбра отображают отношения.

SPARQL - язык запросов к данным, представленным по модели RDF, а также протокол для передачи этих запросов и ответов на них. SPARQL является рекомендацией консорциума W3C и одной из технологий семантической паутины. Предоставление SPARQL-точек доступа (англ. SPARQL-endpoint) является рекомендованной практикой при публикации данных во всемирной паутине.

Необходимо остановиться подробнее на том, что представляют собой информационные системы, построенные на связанных данных. Прежде всего, эти решения относятся к сфере Semantic Web.

Semantic Web – это движение, возникшее благодаря энтузиастам из консорциума W3C. Целью данного движения является привнесение семантики в мировую паутину. По сути, Semantic Web – это преобразование неструктурированного контента Интернета в структурированную паутину данных, несущих смысл. Набор технологий семантической паутины опирается на стандарт RDF – Resource Description Framework - фреймворк описания ресурсов. Как пишет W3C, семантическая паутина обеспечивает общий фреймворк, который позволяет делиться данными, повторно их использовать между приложениями, организациями и сообществами [126].

Итак, семантик веб – это видение того, к чему должен эволюционировать Интернет. Но консорциум не только старается заразить нас своим видением, он также предоставляет нам набор стандартов, а именно: SPARQL, RDF, краткая справка о которых уже приведена выше.

Помимо этих стандартов необходимо уделить внимание стандарту OWL. OWL – web ontology language (язык веб онтологий). Язык предназначен для описания модели данных, то есть классы и взаимоотношения между ними.

Теперь хотелось бы обратить внимание на то, какие возможности дают все эти технологии, когда они будут работать вместе.

Открытая и машиночитаемая модель данных позволяет использовать ее повторно и делает ее прозрачной для поддержки и доработки. Более того, существующие инструменты работы с OWL дадут возможность экспертам в предметных областях понимать и изменять модели данных, не прибегая к помощи программистов. Благодаря открытости и семантической прозрачности модели данных станет возможным создавать обобщенные запросы к данным сразу

нескольких систем, а это в свою очередь позволит стереть границы между системами.

Данные, поступающие благодаря стандарту RDF в тройках (триплетах) дадут возможность обогатить данные семантикой в момент их формирования до отправки в единый центр обработки. Таким образом, это поможет справляться с проблемой разнородности данных.

Так как данные, хранимые и передаваемые в тройках (субъект, объект, предикат) уже предполагают наличие двух вершин и связи между ними, то вся совокупность RDF-данных представляет собой огромный граф, что, в свою очередь, располагает к графическому представлению этих данных и нахождению новых взаимосвязей между ними.

Графическое представление облегчит понимание уже выявленных и заложенных в модель тенденций в данных, а выполнению задачи нахождения новых связей будет способствовать наличие механизмов построения автоматизированных семантических выводов (ризонинг). Под ризонером мы понимаем систему логического вывода в семантических сетях. Ризонер способен на основе уже существующей структуры данных выводить на основе формальной логики новые взаимосвязи [127].

Достоинства:

- а) Преодоление проблемы неструктурированных данных
- б) Открытость модели данных и самих данных и готовность их к использованию в сторонних системах благодаря предзаданной семантике
- в) Возможность упрощенной интеграции сторонних систем как дополнительных источников данных
- г) Быстрая адаптация к меняющейся предметной области

- д) Возможность нахождения дополнительных, новых зависимостей между фактами, событиями, данными, достигаемая за счет применения ризонеров.

Недостатки:

- а) Сравнительно дорогие по времени операции по запросу данных и их обработке
- б) Технологии все еще находятся не на промышленной стадии развития, а лишь на исследовательской и околонаучной.

В целях выполнения ПНИ будет целесообразно и перспективно использовать данный набор технологий ввиду их значительных преимуществ и сулимых ими выгод, но при составлении архитектурного видения системы следует сосредоточиться на преодолении сторон семантического подхода.

## **5.3 Проблема разнородности данных, поступающих от электронных потребительских устройств**

### **5.3.1 Синхронность и асинхронность**

После сравнения различных подходов к решению проблемы Big data, которая будет стоять и в рамках текущего ПНИ и выявления наиболее весомого по своим достоинствам, имеет смысл описать то, как выбранный подход будет справляться со специфическими трудностями, которые будут присущи именно системе сбора и обработки данных с сенсоров в рамках концепции интернета вещей.

Итак, первая проблема сформулирована в заголовке раздела. Данная проблема, свойственная Big Data в нашем случае преломляется в проблеме синхронности и асинхронность взаимодействия.

Как известно, асинхронность добавляет сложности процессу взаимодействия, но информационные системы давно научились справляться с этой проблемой [128]. Но там, где данные могут поступать не только синхронно и асинхронно, но и еще в

больших объемах до сих пор возникают проблемы при реализации. В нашем случае, данные с датчиков будут поступать как в синхронном, так и в асинхронном режимах, что будет усложнять их обработку в реальном времени. При обработке асинхронных данных придется каким-то образом восстанавливать время генерации измерения и приводить его к единому серверному времени. Эти дополнительные операции усложняют обработку в реальном времени и создадут вероятность для возникновения путаницы со временем. Необходимо будет проводить регулярно процедуру корректировки времени на всех устройствах, которые включают значения времени в свои данные.

### **5.3.2 Различные форматы**

Формат поступающих от датчиков данных будет сильно различаться в зависимости от типа устройства, его возможностей предобработки, сложности проводимых им измерений, количеством измеряемых параметров и т.д. Для обработки различных форматов понадобится приводить данные к одним единицам измерения, разрешать противоречивость из-за нестыковок с разрядностью значений. Если же формат представления данных будет составным и представлять собой результаты измерений сразу нескольких параметров, которые имеют между собой какие-либо зависимости. Для решения этой проблемы понадобится заносить в онтологию эти связи и производить разбор сложных пакетов данных.

### **5.3.3 Различная дискретность и задержки**

Проблема различий в дискретности поступления показаний измерительных приборов будет приводить к адаптации алгоритмов обработки под дискретность приборов. Например, если показания счетчика А являются решающим для вывода Б и срабатывания предупреждения В, то необходимо дождаться поступления показаний от счетчика А, но в то же время, какой необходимо установить тайм аут для определения недоступности счетчика? Для адекватного и управляемого решения этой проблемы потребуется прозрачность и читаемость принципов работы алгоритмов и правил для их настройки под конкретный набор датчиков.

### **5.3.4 Применимость технологий связанных данных (Linked Data) для решения проблемы**

Наличие двух видов взаимодействия: синхронного и асинхронного, по-видимому, приведет к необходимости включения в архитектуру модуля обработки очередей сообщений. Как уже писалось необходимо выстраивание всех событий в едином хронологическом порядке вне зависимости от времени их поступления в анализатор. Но в рамках связанных данных появится возможность хранить статические данные об особенностях того или иного измерительного средства, которые нужны для восстановления правильной хронологии событий. Более того, семантические данные делают прозрачным механизм записи алгоритма восстановления хронологии событий, что удешевляет его создание, доработки и поддержку.

По аналогии с предыдущим, правила приведения к унифицированным данным могут быть выражены наглядно в онтологии и, таким образом, обойдется дешевле в разработке и поддержке. Помимо этого, этими правилами можно будет делиться, они могут быть повторно использованы в смежных системах. Если же датчик будет позволять производить предобработку данных, то часть операций может быть вынесена на сторону датчика, например, датчик может маркировать данные таким образом, чтобы указывать алгоритму преобразования данных на запуск того или иного сценария.

## **5.4 Проблема постоянно меняющейся среды**

### **5.4.1 Нормативно-правовое поле**

Ни для кого не секрет, что нормативно-правовое подлежит постоянному изменению. Эти изменения продиктованы сменой технологий, сменой общественных ориентиров, политики государства и т.д. Изменения в правовом поле являются обязательными для всех вовлеченных сторон и информационные системы должны иметь возможность быстро и дешево адаптироваться к постоянно

меняющемуся правовому полю. В отношении семантических технологий можно утверждать, что требования к изменениям могут быть автоматически получены из открытых государственных источников и автоматически же эти требования будут транслированы системе. Этот процесс может проистекать без участия человека и практически не требовать написания кода или вмешательства в работу системы. Таким образом, адаптация системы под меняющиеся правила и регламенты будет происходить дешево, прослеживаемо, оперативно и без остановки работы системы, что особенно ценно для систем, работающих с потоковыми данными.

#### **5.4.2 Смена технологий, форматов, аппаратного обеспечения**

В обычных системах смена технологий, форматов и аппаратного обеспечения зачастую приводит к полному переписыванию всей системы. Не нужно говорить, что разработка новой системы, поддерживающей сменившееся оборудование или форматы затратно по всем видам ресурсов.

В системах, построенных на семантических технологиях при смене оборудования (датчика или шлюза) необходимо только адаптировать характеризующую его часть онтологии, а также правила обработки поступающих от него данных, что не является критичным изменением для системы в целом. Конечно, могут возникать ситуации, когда внешняя среда будет выдвигать требования по производительности, которым система уже не будет способна удовлетворить, но системы, построенные на семантических технологиях способны выдержать гораздо более значительные внешние изменения, нежели традиционные системы [158].

Подчас в традиционных системах набор технологий и фреймворков становится несовместим с новыми требованиями оборудования.

#### **5.4.3 Запросы пользователей и новый функционал**

Что касается свойств адаптивности системы к нуждам пользователя, то для этой задачи семантические технологии тоже являются наиболее предпочтительным решением. Внедренная в данные семантика и свойство данных быть понятными для

программ позволяет персонализировать эти данные под предпочтения конкретного пользователя, под модели его поведения, под его явно выраженные интересы в пользовательского профиле. В данном случае, будет решаться задача совмещения предпочтений, преломленных на концепты онтологии и семантики данных. Алгоритм будет в реальном времени ранжировать содержимое по степени соответствия интересам пользователя [130]. Системы с подобным уровнем интеллектуальности уже существуют и успешно функционируют в сети Интернет, например, примером такой системы может выступать новостной портал британской вещательной компании – BBC<sup>1</sup>. Он построен по описанному выше принципу и способен адаптировать новостной поток под нужды конкретного посетителя в зависимости от явно представленных и неявно выраженных через поведение пользователя на портале предпочтений.

Могут ли семантические технологии упростить задачу создания новых функций системы? По утверждению экспертов из W3C семантические технологии действительно способны облегчить разработку новых функций при условии, что обновляемая система уже имеет в себе семантические ядро. Благодаря тому, что все предметное знание и знание о принципах работы системы находится в форме онтологии, можно быть уверенным, что будет соблюдена целостность и последовательность модели предметной области и новые функции не войдут в конфликт с ранее разработанными. Семантическая паутина может быть применена на уровне взаимодействия модулей системы и явно выраженные интерфейсы взаимодействия облегчат создания новых модулей, которые будут встраивать в существующую инфраструктуру [160].

## 5.5 Решение проблемы обработки данных в реальном времени

Несмотря на кажущуюся неповоротливость систем, построенных на семантических технологиях индустрия уже выработала механизмы обработки поточных данных в реальном времени. Например, проведены успешные

---

<sup>1</sup> BBC, URL: <http://bbc.com/> (дата обращение: 30.12.2014)

эксперименты обработки больших массивов поточных разнородных данных на уровне города Дублин [132]. Существуют технологии, которые позволяют формировать SPARQL-запросы к поточным данным и выполнять их в реальном времени. При этом возможно создать систему правил, которая бы не только обрабатывала поточные данные, но и обращалась к уже сохраненным данным. Такая функциональность достаточна для работы системы реального времени по оперативной обработке данных и выявлению ситуаций, отклоняющихся от штатного сценария [133]. Самым эффективным подходом при обработке больших объемов поточных данных можно назвать тот, при котором данные обрабатываются в оперативной памяти без сохранения на диск. Только при срабатывании каких-то отдельных правил событий и их контекст сохраняются на диск [163]. Такое выключение из процесса обработки процедуры записи на диск существенно ускоряет весь процесс, потому что действие записи на жесткие диски все-таки по сей день является затратным по времени.

## **5.6 Решение задачи визуализации больших массивов данных в современных платформах**

Ранее мы показали потенциал семантических технологий в решении проблемы Big Data, продемонстрировали выгоды от использования семантических технологий при построении систем в сфере интернета вещей. Теперь же хотелось бы остановиться на аспектах использования семантических технологий в задачах, связанных с визуализацией больших объемов данных. Каковы будут преимущества семантических технологий? И не будет ли сравнительная новизна семантических технологий препятствовать внедрению их в решения по визуализации данных? На эти вопросы мы дадим ответы ниже, но для начала рассмотрим типовые решения, нацеленные на решения задачи наглядного представления значительных объемов данных.

Прежде всего, задача визуализации распадается на две стадии:

- a) Предварительная обработка данных

б) Непосредственное выводение на экран форм представления данных (рендеринг)

Предварительная обработка данных выполняется на поточных данных, и, если мы говорим о семантических технологиях, то, по-видимому, эти данные будут представлены в RDF-потоке. Для задания контекста поступающим данным они будут обогащаться за счет знаний предметной области, хранящихся в онтологии. После чего будут вступать в работу правила обработки данных, которые будут также иметь возможность обращаться к хранилищу и сравнивать поточные данные с имеющимися сохраненными фактами и данными [135].

Срабатывание правил, во-первых, приводит к сохранению событий в архив (хранилище). Во-вторых, события могут инициировать визуализацию каких-либо процессов или значений. Правила также управляют подпиской тех или иных механизмов визуализации на поточные данные. Допустим, график давления может быть подписан только на показания только датчиков Н-34 и Н-38, находящихся в котельной и за перенаправление этих данных на конкретный график, визуализируемый в конкретной точке системы, отвечает соответствующее правило в блоке поточного анализа.

Основой описанного выше архитектурного решения служат связанные поточные данные (Linked Stream Data – LSD), которые, в свою очередь, представляют собой поточный RDF, обогащенный временными данными и контекстом из онтологии предметной области. Остановимся на преимуществах применения LSD в качестве формы существования данных системы.

Наиболее важное преимущество – это наличие дополнительные сведений, содержащихся в поточном RDF. Эти дополнительные сведения на стадии обработки ускорят ее, зададут верный сценарий визуализации и предрешат срабатывание правил. За счет онтологии предметной области будет возможным получить контекст, в котором следует рассматривать получаемые данные. Например, помимо показаний датчика #775 можно сказать, что находится в машинном отсеке, его

показания влияют на работу двигателя и т.д. Эти сведения очень существенны для интерпретации показаний.

Важно отметить еще и тот аспект, что LSD являются машиночитаемыми данными и могут быть переданы в сторонние системы, а также их обработку можно легко и дешево распределить между подсистемами или различными географическими локациями.

В части дальнейшей визуализации LSD также характеризуются некоторыми преимуществами, а именно: сама графическая форма существования данных в виде триплетов предполагает упрощенное построение графов, наглядно показывающих зависимость одних данных от других.

Наконец, нельзя забывать о сильных сторонах инструментов логического вывода. Эти инструменты постоянно развиваются, предоставляя все большие возможности по автоматическому нахождения новых связей между данными, что служит основой для выявления новых глобальных взаимосвязей между явлениями и событиями, которые не могут быть выявлены человеком в огромных массивах данных.

## ЗАКЛЮЧЕНИЕ

В отчете представлены результаты работ по выбору направления исследований по 1 этапу ПНИ «Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)»

В данном отчете представлены следующие работы, в соответствии с Планом графиком работ (далее, ПГ):

- а) Аналитический обзор современной научно-технической, нормативной, методической литературы, затрагивающей научно-техническую проблему, исследуемую в рамках ПНИ, в том числе обзор научных информационных источников: статьи в ведущих зарубежных и (или) российских научных журналах, монографии и (или) патенты) - не менее 15 научно-информационных источников за период 2009 - 2014 гг
- б) Патентные исследования в соответствии с ГОСТ Р 15.011-96
- в) Выбор направления исследований в области разработки масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)
- г) Разработка ПАС с подсистемой виртуальных серверов, подсистемой интеграции больших массивов гетерогенных данных и симуляцией

распределенных вычислений для моделирования сети электронных потребительских устройств (Internet of Things)

- д) Анализ существующих приемов онтологического инжиниринга и технологий семантического веба для агрегации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных
- е) Анализ применимости технологии связанных данных (Linked Data) для задачи интерактивной визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных

Основные результаты, полученные за отчетный период и описаные в данном отчете о ПНИ:

- а) В рамках работы по п.1.1 ПГ проведен аналитический обзор современной научно-технической, нормативной, методической литературы, затрагивающей научно-техническую проблему, исследуемую в рамках ПНИ, в количестве 41-го научно-информационного источника, из которых:
  - 1) 28 источников за период 2009-2014 гг.
  - 2) 13 источников за период до 2009 г.

Было выполнено требование ТЗ по количеству научно-информационных источников используемых в аналитическом обзоре, не менее 15 за период 2009-2014 гг.

- б) В рамках работы по п.1.2 ПГ проведены патентные исследования, в рамках которых было найдено 45 патентов для их классификации и последующего анализа, из которых было выделено 7 патентов, относящихся не постредственно к научно-технической проблеме, исследуемой в рамках ПНИ.

Проведенные патентные исследования соответствуют требованиям ТЗ и включены в отчет о ПНИ. Также на следующих этапах ПНИ при

получении результатов интеллектуальной деятельности, способных к правовой охране, будут проведены дополнительные патентные исследования в соответствии с ГОСТ Р 15.011-96.

Также по результатом патентных исследований было выявлено, что планируемые к получению в рамках ПНИ научно-технические результаты могут пригодными к правовой охране.

в) В рамках работы по п.1.3 ПГ и на основе результатов работы по п.1.1 ПГ были выбраны следующие направления исследований в рамках данного ПНИ:

- 1) методы и алгоритмы агрегации больших массивов гетерогенных данных, в том числе протоколы передачи данных прикладного уровня
- 2) методы и алгоритмы нормализации и анализа больших массивов гетерогенных данных генерируемых распределенной сетью электронных потребительских устройств, в том числе моделирование данных с помощью онтологий
- 3) методы и алгоритмы интерактивной визуализации больших массивов гетерогенных данных
- 4) методы передачи данных от потребительских устройств к центрам обработки данных
- 5) технические принципы и подходы к организации защиты информации передаваемой в распределенной сети электронных потребительских устройств
- 6) Так же в отчете о ПНИ приведены доводы свидетельствующие об актуальности выбранных направлений исследований.

г) В рамках работы по п.1.4 ПГ был разработан программно-аппаратный стенд (далее - ПАС) с подсистемой виртуальных серверов, подсистемой интеграции больших массивов гетерогенных данных и симуляцией распределенных вычислений для моделирования сети электронных потребительских устройств.

ПАС обеспечивает симуляцию распределенной сети потребительских устройств со следующими характеристиками:

- 1) количество электронных потребительских устройств до 2000
- 2) с частотой обновления текущих показаний каждым из электронных потребительских устройств до 1 показания/15 минут

При разработке ПАС были использованы современные протоколы передачи данных от электронных потребительских устройств, которые имеют низкие требования к вычислительным ресурсам и предоставляют дополнительные возможности для работы с распределенной сетью электронных потребительских устройств:

- 1) уникальная идентификация устройств в сети Интернет по средствам URI-адресов
- 2) совместимость протоколов с протоколом HTTP
- 3) обнаружение устройств (ресурсов) в сети и автоматическая идентификация их возможностей
- 4) обеспечение инициализации обмена сообщениями со стороны устройств, а не их клиентов

Также для обеспечения синтаксической и семантической интероперабельности, были использованы существующие онтологии, например, Semantic Sensor Ontology, и разработаны новые верхнеуровневые онтологии.

В отчете о ПНИ представлена эскизная конструкторская документация на ПАС в составе:

- 1) схемы структурной в соответствии с ГОСТ 2.701-84
- 2) инструкции по эксплуатации
- 3) формуляра в соответствии с ГОСТ 2.601-2006 и ГОСТ 2.610-2006

Были достигнуты технические требования ТЗ на ПАС, требования на разрабатываемую документацию и разработанный ПАС готов к использованию на последующих этапах ПНИ.

д) В рамках работы по п.1.5 ПГ был проведен анализ существующих приемов онтологического инжиниринга и технологий семантического веба для агрегации больших массивов гетерогенных данных. Представлена типовая архитектура программного обеспечения, обрабатывающего данные распределенной сети электронных потребительских устройств, которая характеризуется:

- 1) изменчивостью и гетерогенностью данных, устройств и сетей
- 2) использованием разнообразных источников данных
- 3) наличием нескольких административных зон
- 4) необходимостью в управлении несколькими, конкурентными и не скоординированными запросами к такой распределенной сети

В результате анализа были выявлены следующие характеристики существующих приемов для агрегации гетерогенных данных, которые могут быть улучшены в рамках данного ПНИ:

- 1) плохая масштабируемость решений для первичного сбора гетерогенных данных
- 2) высокие требования к вычислительным ресурсам электронных потребительских устройств

Информация и материалы, полученные в рамках данного анализа, будут использованы на последующих этапах ПНИ для выполнения работ по ТЗ, в частности для получения результатов перечисленных в пп. 2.2, 2.3, 2.5.

е) В рамках работы по п.1.6 ПГ был проведен анализ применимости технологии связанных данных для задачи интерактивной визуализации больших массивов гетерогенных данных.

Выявлены существующие проблемы визуализации больших массивов гетерогенных данных:

- 1) синхронность и асинхронность поступления данных
- 2) различие форматов данных
- 3) различная дискретность и задержки в поступлении данных

А так же предложены решения перечисленных выше проблем, такие как:

- 1) унификация схем данных с помощью верхнеуровневых онтологий
- 2) обогащение временных (потоковых) данных поступающих с распределенной сети электронных потребительских устройств
- 3) логический вывод на существующих онтологических данных

Информация и материалы, полученные в рамках данного анализа, будут использованы на последующих этапах ПНИ для выполнения работ по ТЗ, в частности для получения результатов перечисленных в пп. 2.1-2.3.

На отчетном этапе были достигнуты все необходимые результаты, предусмотренные ПГ, которые удовлетворяют требования, сформулированным в ТЗ. Команда проекта проявила способность оперативноправляться с исследовательскими и техническими задачами и продемонстрировала готовность к выполнению серьезных прикладных научно-технических разработок.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Ashton Kevin.* The Internet of Things // RFID Journal. 2009. N 50.
- 2 *Atzori Luigi, Iera Antonio, Morabito Giacomo.* The Internet of Things: A survey // Computer Networks. 2010. Vol. 54, N 15. P. 2787-2805. ISSN 1389-1286.
- 3 *Sundmaeker Harald, Guillemin Patrick, Friess Peter, Woelfflé Sylvie.* Vision and Challenges for Realising the Internet of Things. 2010. ISBN 978-92-79-15088-3.
- 4 *Belissent J.* Getting clever about smart cities: new opportunities require new. Forester Research. 2010.
- 5 *Gubbi Jayavardhana, Buyya Rajkumar, Marusic Slaven, Palaniswami Marimuthu.* Internet of Things (IoT): A vision, architectural elements, and future directions // Future Generation Computer Systems. 2013. Vol. 29. P. 1645-1660.
- 6 *Welbourne E, Battle L, Cole G, Gould K, Rector K, Raymer S, Balazinska M, Borriello G.* Building the Internet of Things Using RFID: The RFID Ecosystem Experience // IEEE Internet Computing. 2009. Vol. 13, N 3. P. 48-55.
- 7 *Juels A.* RFID security and privacy: a research survey // IEEE Journal on Selected Areas in Communications. 2006. Vol. 24, N 2. P. 381-394.
- 8 *Akyildiz I F, Su W, Sankarasubramaniam Y, Cayirci E.* Wireless sensor networks: a survey // Computer Networks. 2002. Vol. 38, N 4. P. 393 - 422.
- 9 *Amitabha Ghosh, Sajal Das.* Coverage and connectivity issues in wireless sensor networks: A survey // Pervasive and Mobile Computing. 2008. Vol. 4, N 3. P. 303-334.
- 10 *Kumar Jha Mukesh, Sharma T P.* Secure Data aggregation in Wireless Sensor Network: A Survey // International Journal of Engineering Science and Technology. 2011. Vol. 3, N 3. P. 2013-2019.

- 11 *Zorzi M, Gluhak A, Lange S, Bassi A.* From today's INTRANet of things to a future INTERnet of things: a wireless- and mobility-related view // IEEE Wireless Communications. 2010. Vol. 17, N 6. P. 44-51.
- 12 *Honle N, Kappeler U, Nicklas Daniela, Schwarz T, Grossmann M.* Benefits of Integrating Meta Data into a Context Model // Third IEEE International Conference on Pervasive Computing and Communications Workshops. 2005. P. 25-29.
- 13 Intel Unifies and Simplifies Connectivity, Security for IoT [Electronic resource] // Intel Newsroom. 2014. Mode of access : [http://newsroom.intel.com/community/intel\\_newsroom/blog/2014/12/09/intel-unifies-and-simplifies-connectivity-security-for-iot](http://newsroom.intel.com/community/intel_newsroom/blog/2014/12/09/intel-unifies-and-simplifies-connectivity-security-for-iot) (30.12.2014).
- 14 *Miorandi Daniele, Sicari Sabrina, De Pellegrini Francesco, Chlamtac Imrich.* Internet of things: Vision, applications and research challenges // Ad Hoc Networks. 2012. Vol. 10, N 7. P. 1497-1516.
- 15 *Clarke James, Castro Rodrigo Roman, Sharma Abhishek, Lopez Javier, Suri Neeraj.* Trust & Security RTD in the Internet of Things: Opportunities for International Cooperation // Proceedings of the First International Conference on Security of Internet of Things. Kollam, India : ACM, 2012. P. 172--178.
- 16 *Xu Li Da.* Introduction: Systems Science in Industrial Sectors // Systems Research and Behavioral Science. 2013. Vol. 30, N 3. P. 211--213.
- 17 *Li Fachao, Jin Chenxia, Jing Ying, Wilamowska-Korsak Marzena, Bi Zhuming.* A Rough Programming Model Based on the Greatest Compatible Classes and Synthesis Effect // Systems Research and Behavioral Science. 2013. Vol. 30, N 3. P. 229-243.
- 18 *Li Da Xu.* Enterprise Systems: State-of-the-Art and Future Trends // IEEE Transactions on Industrial Informatics. 2011. Vol. 7, N 4. P. 630 - 640. ISSN 1551-3203.

- 19 Andreini F, Crisciani F, Cicconetti C, Mambrini R. Context-aware location in the Internet of Things // IEEE GLOBECOM Workshops. Miami, FL : IEEE, 2010. P. 300 - 304. ISBN 978-1-4244-8863-6.
- 20 Pujolle G. An Autonomic-oriented Architecture for the Internet of Things // IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing. Sofia : IEE, 2006. P. 163 - 168. ISBN 0-7695-2643-8.
- 21 Gronbaek I. Architecture for the Internet of Things (IoT): API and Interconnect // Second International Conference on Sensor Technologies and Applications. Cap Esterel : IEEE, 2008. P. 802 - 807. ISBN 978-0-7695-3330-8.
- 22 James Anne, Cooper Joshua, Jeffery Keith, Saake Gunter. Research Directions in Database Architectures for the Internet of Things: A Communication of the First International Workshop on Database Architectures for the Internet of Things // Lecture Notes in Computer Science. 2009. Vol. 5588. P. 225-233. ISSN 0302-9743.
- 23 Spiess P, Karnouskos S, Guinard D, Savio D, Baecker O, Souza L, Trifa V. SOA-Based Integration of the Internet of Things in Enterprise Services // IEEE International Conference on Web Services. Los Angeles, CA : IEEE, 2009. P. 968 - 975. ISBN 978-0-7695-3709-2.
- 24 Castellani A P, Bui N, Casari P, Rossi M, Shelby Z, Zorzi M. Architecture and protocols for the Internet of Things: A case study // IEEE International Conference on Pervasive Computing and Communications Workshops. Mannheim : IEEE, 2010. P. 678 - 683. ISBN 978-1-4244-6605-4.
- 25 Castellani A P, Gheda M, Bui N, Rossi M, Zorzi M. Web Services for the Internet of Things through CoAP and EXI // IEEE International Conference on Communications Workshops. Kyoto : IEEE, 2011. P. 1-6. ISBN 978-1-61284-954-6.
- 26 Guinard Dominique, Trifa Vlad, Mattern Friedemann, Wilde Erik. Architecting the Internet of Things. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19156-5.

- 27 *Giner Pau, Cetina Carlos, Fons Joan, Pelechano Vicente.* Developing Mobile Workflow Support in the Internet of Things // IEEE Pervasive Computing. 2010. Vol. 9, N 2. P. 18-26.
- 28 *Kawsar F, Kortuem G, Altakrouri B.* Supporting interaction with the Internet of Things across objects, time and space // Internet of Things. Tokyo : IEEE, 2010. P. 1-8. ISBN 978-1-4244-7413-4.
- 29 *Främling Kary, Nyman Jan.* Information architecture for intelligent products in the internet of things // Proceedings of NOFOMA. 2008. P. 224-229.
- 30 *Kortuem G, Kawsar F, Fitton D, Sundramoorthy V.* Smart objects as building blocks for the Internet of things // IEEE Internet Computing. 2009. Vol. 14, N 1. ISSN 1089-7801.
- 31 *Perera C, Jayaraman P, Zaslavsky A, Christen P, Georgakopoulos D.* Dynamic configuration of sensors using mobile sensor hub in internet of things paradigm // IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing. Melbourne, VIC : IEEE, 2013. P. 473 - 478. ISBN 978-1-4673-5499-8.
- 32 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats [Electronic resource] // IEEE Xplore. 2007. Mode of access : <http://dx.doi.org/10.1109/IEEESTD.2007.4346346> (30.12.2014).
- 33 *Botts Mike.* OpenGIS® Sensor Model Language (SensorML) Implementation Specification [Electronic resource] // 2005. Mode of access : [https://portal.opengeospatial.org/modules/admin/license\\_agreement.php?suppressHeaders=0&access\\_license\\_id=3&target=http://portal.opengeospatial.org/files/%3fartifact\\_id=12606](https://portal.opengeospatial.org/modules/admin/license_agreement.php?suppressHeaders=0&access_license_id=3&target=http://portal.opengeospatial.org/files/%3fartifact_id=12606) (30.12.2014).
- 34 *Compton M, Hensori C A, Neuhaus H, Lefort L, Sheth Amit.* A Survey of the Semantic Specification of Sensors // International Workshop on Semantic Sensor Networks. 2009.

- 35 *Perera C, Zaslavsky A, Salehi A, Georgakopoulos D.* Connecting mobile things to global sensor network middleware using system-generated wrappers // Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access. s.l. : ACM, 2012. P. 23-30. ISBN 978-1-4503-1442-8.
- 36 *Wang Feng, Ge Baoshan, Zhang Li, Chen Yong, Yang Xin, Li Xiayuan.* A System Framework of Security Management in Enterprise Systems // Systems Research and Behavioral Science. 2013. Vol. 30, N 3. P. 287–299.
- 37 *Katasonov A, Kaykova O, Khriyenko O, Nikitin S, Terziyan V.* Smart Semantic Middleware for the Internet of Things // Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics, Intelligent Control Systems and Optimization. Madeira, Portugal : INSTICC Press, 2008. P. 169-178. ISBN 978-989-8111-30-2.
- 38 *Caporuscio M, Raverdy P, Issarny V.* ubiSOAP: A Service-Oriented Middleware for Ubiquitous Networking // IEEE Transactions on Services Computing. 2010. Vol. 5, N 1. P. 86 - 98. ISSN 1939-1374.
- 39 *Wang X H, Qing Zhang Da, Tao Gu, Pung H K.* Ontology based context modeling and reasoning using OWL // Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. s.l. : IEEE, 2014. P. 18 - 22. ISBN 0-7695-2106-1.
- 40 *Emmanuel Spanos Dimitrios, Stavrou Periklis, Mitrou Nikolas, Konstantinou Nikolas.* SensorStream: a semantic real-time stream management system // International Journal of Ad Hoc and Ubiquitous Computing. 2012. Vol. 11, N 2. ISBN 1743-8225.
- 41 *Heath Tom, Bizer Christian.* Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool, 2011. ISBN 9781608454303.
- 42 *Phuoc Danh Le, Hauswirth Manfred.* Linked Open Data in Sensor Data Mashups // Proceedings of the 2nd International Workshop on Semantic Sensor Networks. 2009.

- 43 *Perera C, Zaslavsky A, Christen P, Georgakopoulos D.* CA4IOT: Context Awareness for Internet of Things // IEEE International Conference on Green Computing and Communications. Besancon : IEEE, 2012. P. 775 - 782. ISBN 978-1-4673-5146-1.
- 44 *Perera C, Zaslavsky A, Compton M, Georgakopoulos D.* Context-Aware Sensor Search, Selection and Ranking Model for Internet of Things Middleware // IEEE 14th International Conference on Mobile Data Management. Milan : IEEE, 2013. P. 314 - 322. ISBN 978-1-4673-6068-5.
- 45 *Maximilien E M, Singh M P.* A framework and ontology for dynamic Web services selection // IEEE Internet Computing. 2004. Vol. 8, N 5. P. 84 - 93. ISSN 1089-7801.
- 46 *Shuping Ran.* A Model for Web Services Discovery with QoS // SIGecom Exch.. 2003. Vol. 4, N 1. ISSN 1551-9031.
- 47 *Bandyopadhyay Debasis, Sen Jaydip.* Internet of things: applications and challenges in technology and standardization // Wireless Personal Communications. 2011. Vol. 58, N 1. P. 49-69.
- 48 *Roman R, Najera P, Lopez J.* Securing the Internet of Things // Computer. 2011. Vol. 44, N 9. P. 51-58. ISSN 0018-9162.
- 49 *Tao Yan , Qiaoyan Wen.* A Trust-Third-Party Based Key Management Protocol for Secure Mobile RFID Service Based on the Internet of Things // Advances in Intelligent and Soft Computing. 2012. Vol. 135. P. 201-208. ISBN 978-3-642-27707-8.
- 50 *Mahalle Parikshit, Sachin Babar, Neeli Prasad, Ramjee Prasad.* Identity Management Framework towards Internet of Things (IoT): Roadmap and Key Challenges // Communications in Computer and Information Science. 2010. Vol. 89. P. 430-439. ISBN 978-3-642-14477-6.
- 51 *Gluhak A, Krco S, Nati M, Pfisterer D, Mitton N, Razafindralambo T.* A survey on facilities for experimental internet of things research // Communications Magazine, IEEE. 2011. Vol. 49, N 11. P. 58-67. 0163-6804.

- 52 *Luo Haiyan, Ci Song, Wu Dalei , Stergiou N, Siu Ka-Chun* . A remote markerless human gait tracking for e-healthcare based on content-aware wireless multimedia communications // Wireless Communications, IEEE. 2010. Vol. 17, N 1. P. 44-50. ISSN 1536-1284.
- 53 *Nussbaum Gerhard*. People with Disabilities: Assistive Homes and Environments // Computers Helping People with Special Needs. 2006. Vol. 4061. P. 457-460. ISBN 978-3-540-36020-9.
- 54 *Alkar A Z, Buhur U*. An Internet based wireless home automation system for multifunctional devices // IEEE Transactions on Consumer Electronics. 2005. Vol. 51, N 4. P. 1169-1174. ISSN 0098-3063.
- 55 *Darianian M, Michael M P*. Smart Home Mobile RFID-Based Internet-of-Things Systems and Services // International Conference on Advanced Computer Theory and Engineering. 2008. P. 116-120.
- 56 *Huansheng Ning, Ziou Wang*. Future Internet of Things Architecture: Like Mankind Neural System or Social Organization Framework? // IEEE Communications Letters. 2011. Vol. 15, N 4. P. 461-463. ISSN 1089-7798.
- 57 *Atzori L, Iera A, Morabito G*. SIoT: Giving a Social Structure to the Internet of Things // IEEE Communications Letters. 2011. Vol. 15, N 11. P. 1193-1195. ISSN 1089-7798.
- 58 *Xu Li, Rongxing Lu, Xiaohui Liang, Xuemin Shen, Jiming Chen, Xiaodong Lin*. Smart community: an internet of things application // IEEE Communications Magazine. 2011. Vol. 49, N 11. P. 68-75. ISSN 0163-6804.
- 59 *Cory Kidd, Orr Robert, Abowd Gregory, G. Atkeson Christopher, Essa Irfan, MacIntyre Blair , Mynatt Elizabeth , Starner Thad, Newstetter Wendy* . The Aware Home: A Living Laboratory for Ubiquitous Computing Research // Cooperative Buildings. Integrating Information, Organizations, and Architecture. 1999. Vol. 1670. P. 191-198.

- 60 Future Retail Center [Electronic resource] // SAP Living Labs. Mode of access : <http://www.sap.com/corporate-en/about/our-company/innovation/living-labs/future-retail.html> (29.12.2014).
- 61 *Hernández-Muñoz Jose, Bernat Vercher Jesús , Muñoz Luis , Galache José, Presser Mirko, Hernández Gómez Luis, Pettersson Jan* . Smart Cities at the Forefront of the Future Internet // The Future Internet. 2011. Vol. 6656. P. 447-462. ISBN 978-3-642-20897-3.
- 62 *Murty R N, Mainland G, Rose I, Chowdhury A R, Gosain A, Bers J, Welsh M*. CitySense: An Urban-Scale Wireless Sensor Network and Testbed // IEEE Conference on Technologies for Homeland Security. 2008. P. 583-588.
- 63 *Liu D L, Zhu X B, Xu K L, Fang D M*. The Design of Modern Agriculture Control System Based on Internet of Things // Applied Mechanics and Materials. 2014. Vol. 517. P. 1519-1522.
- 64 *Bainbridge Scott, Steinberg C, Furnas M*. GBROOS—an ocean observing system for the Great Barrier Reef // International Coral Reef Symposium. 2010. P. 529-533.
- 65 *Johnstone Ron, Caputo Davide, Cella Umberto , Gandelli Alessandro , Alippi Cesare , Grimaccia Francesco , Haritos N, Zich Riccardo*. Smart Environmental Measurement & Analysis Technologies (SEMAT): Wireless sensor networks in the marine environment // Proceedings of the Wireless Sensor and Actuator Network Research on Opposite Sides of the Globe. 2008.
- 66 *Zhang Min , Yu Tao , Fang Zhai Guo* . Smart Transport System Based on “The Internet of Things” // Applied Mechanics and Materials. 2011. Vol. 49. P. 1073-1076.
- 67 *Hong-En Lin, Zito Rocco , Taylor M*. A review of travel-time prediction in transport and logistics // Proceedings of the Eastern Asia Society for transportation studies. 2005. Vol. 5. P. 1433-1448.

- 68 *Miao Yun, Bu Yuxin.* Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid // International Conference on Advances in Energy Engineering. 2010. P. 69-72.
- 69 *Akyildiz Ian, Melodia Tommaso , Chowdhury Kaushik.* A survey on wireless multimedia sensor networks // Computer Networks. 2007. Vol. 51, N 4. P. 921 - 960.
- 70 *Zhao Liqiang, Yin Shouyi, Liu Leibo, Zhen Zhang , Wei Shaojun.* A Crop Monitoring System Based on Wireless Sensor Network // Procedia Environmental Sciences. 2011. Vol. 11. P. 558 - 565.
- 71 *Kumar P, Surendra Ranganath, Huang Weimin, Sengupta K.* Framework for real-time behavior interpretation from traffic video // IEEE Transactions on Intelligent Transportation Systems. 2005. Vol. 6, N 1. P. 43-53.
- 72 *Mayer-Schönberger Viktor.* Failing to Forget the "Drunken Pirate" *Delete: the virtue of forgetting in the digital age.* 2011, pp. 1-16.
- 73 *Ed J. Hui, Thubert P.* Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks [Electronic resource] // The Internet Engineering Task Force. 2011. Mode of access : <https://tools.ietf.org/html/rfc6282> (19.12.2014).
- 74 *Winter T, Thubert P, Brandt A, Hui J, Kelsey R, Levis P, Pister K, Struik R, Vasseur JP, Alexander R.* RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks [Electronic resource] // The Internet Engineering Task Force. 2012. Mode of access : <http://tools.ietf.org/html/rfc6550> (19.12.2014).
- 75 *Dunkels Adam.* Full TCP/IP for 8-bit architectures // Proceedings of the 1st international conference on Mobile systems, applications and services. 2003. P. 85-98.
- 76 *Hui Jonathan, Culler David.* IP is dead, long live IP for wireless sensor networks // Proceedings of the 6th ACM conference on Embedded network sensor systems. 2008. P. 15-28. ISBN 978-1-59593-990-6.

77 UPnP Device Architecture 1.0 [Electronic resource] // UPnP Forum. 2008. Mode of access : <http://www.upnp.org/specs/arch/upnp-arch-devicearchitecture-v1.0.pdf> (19.12.2014).

78 Tolle G. Embedded Binary HTTP (EBHTTP) [Electronic resource] // The Internet Engineering Task Force. 2010. Mode of access : <http://tools.ietf.org/html/draft-tolle-core-ebhttp-00> (19.12.2014).

79 Duquennoy S, Grimaud G, Vandewalle J. Smews: Smart and Mobile Embedded Web Server // International Conference on Complex, Intelligent and Software Intensive Systems. 2009. P. 571-576.

80 Fielding Roy, Taylor Richard. Principled design of the modern Web architecture // ACM Transactions on Internet Technology (TOIT). 2002. Vol. 2, N 2. P. 115-150.

81 Nottingham M. Web Linking [Electronic resource] // The Internet Engineering Task Force. 2010. Mode of access : <https://tools.ietf.org/html/rfc5988> (19.12.2014).

82 Shelby Z. CoRE Link Format [Electronic resource] // The Internet Engineering Task Force. 2012. Mode of access : <https://tools.ietf.org/html/draft-ietf-core-link-format-14> (19.12.2014).

83 Hartke K. Observing Resources in CoAP [Electronic resource] // The Internet Engineering Task Force. 2014. Mode of access : <http://tools.ietf.org/html/draft-ietf-core-observe-14> (19.12.2014).

84 Bormann C, Shelby Z. Blockwise transfers in CoAP [Electronic resource] // The Internet Engineering Task Force. 2011. Mode of access : <http://tools.ietf.org/html/draft-ietf-core-block-03> (19.12.2014).

85 Rahman A, Dijk E. Group Communication for CoAP [Electronic resource] // The Internet Engineering Task Force. 2011. Mode of access : <https://tools.ietf.org/html/draft-rahman-core-groupcomm-07> (19.12.2014).

- 86 *Nottingham M, Hammer-Lahav E.* Defining Well-Known Uniform Resource Identifiers (URIs) [Electronic resource] // The Internet Engineering Task Force. 2010. Mode of access : <https://tools.ietf.org/html/rfc5785> (19.12.2014).
- 87 *Lerche C, Hartke K, Kovatsch M.* Industry adoption of the Internet of Things: A constrained application protocol survey // IEEE 17th Conference on Emerging Technologies Factory Automation (ETFA). 2012. P. 1-6. ISSN 1946-0740.
- 88 *Bormann C.* Guidance for Light-Weight Implementations of the Internet Protocol Suite [Electronic resource] // The Internet Engineering Task Force. 2012. Mode of access : <https://tools.ietf.org/html/draft-bormann-lwig-guidance-01> (19.12.2014).
- 89 *Corcho Oscar, García-Castro Raúl.* Five challenges for the Semantic Sensor Web // Semantic Web. 2010. Vol. 1, N 1-2. P. 121-125.
- 90 *Jeung H, Sarni S, Paparrizos I, Sathe S, Aberer K, Dawes N, Papaioannou T, Lehning M.* Effective Metadata Management in Federated Sensor Networks // IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing. Newport Beach, USA : IEEE, 2010. P. 107-114. ISBN 978-1-4244-7087-7.
- 91 *Motwani Rajeev, Widom Jennifer, Arasu Arvind, Babcock Brian, Babu Shivnath, Datar Mayur, Manku Gurmeet, Olston Chris, Rosenstein Justin, Varma Rohit.* Query Processing, Resource Management, and Approximation in a Data Stream Management System. Stanford InfoLab. Stanford : Stanford InfoLab, 2002.
- 92 *Ahmad Y, Nath S.* COLR-Tree: Communication-Efficient Spatio-Temporal Indexing for a Sensor Data Web Portal // IEEE 24th International Conference on Data Engineering. Cancun : IEEE, 2008. P. 784-793. ISBN 978-1-4244-1836-7.
- 93 *Jian Li, Deshpande A, Khuller S.* Minimizing Communication Cost in Distributed Multi-query Processing // IEEE 25th International Conference on Data Engineering. Shanghai : IEEE, 2009. P. 772 - 783. ISBN 978-1-4244-3422-0.

- 94 *Wu Ji, Zhou Yongluan, Aberer Karl, Tan Kian-Lee.* Towards integrated and efficient scientific sensor data processing: a database approach // Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. New York : s.n., 2009. P. 922-933. ISBN 978-1-60558-422-5.
- 95 *Compton M, Neuhaus H, Taylor K, Tran K.* Reasoning about Sensors and Compositions // International Workshop on Semantic Sensor Networks. 2009.
- 96 *Barbieri Davide, Braga Daniele, Ceri Stefano, Della Valle Emanuele, Grossniklaus Michael.* C-SPARQL: SPARQL for continuous querying // Proceedings of the 18th international conference on World wide web. Madrid, Spain : s.n., 2009.
- 97 *Bröring Arne, Janowicz Krzysztof, Stasch Christoph, Kuhn Werner.* Semantic Challenges for Sensor Plug and Play // Lecture Notes in Computer Science. 2009. Vol. 5886. P. 72-86. ISBN 978-3-642-10600-2.
- 98 *Wei Wang, Barnaghi Payam.* Semantic Annotation and Reasoning for Sensor Data // Lecture Notes in Computer Science. 2009. Vol. 5741. P. 66-76. ISBN 978-3-642-04470-0.
- 99 *Henson C A, Pschorr J K, Sheth A P, Thirunarayan K.* SemSOS: Semantic sensor Observation Service // International Symposium on Collaborative Technologies and Systems. Baltimore, MD : IEE, 2009. P. 44-53. ISBN 978-1-4244-4584-4.
- 100 *Janowicz K, Compton M.* The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology // International Workshop on Semantic Sensor Networks. 2010.
- 101 *Calbimonte Jean-Paul, Jeung Hoyoung, Corcho Oscar, Aberer Karl.* Semantic Sensor Data Search in a Large-Scale Federated Sensor Network // International Workshop on Semantic Sensor Networks. 2011.
- 102 *Berners-Lee Tim.* Linked Data - Design Issues [Electronic resource] // W3C. 2009. Mode of access : <http://www.w3.org/DesignIssues/LinkedData.html> (28.12.2014).

- 103 *Shneiderman B.* The eyes have it: a task by data type taxonomy for information visualizations // IEEE Symposium on Visual Languages. Boulder, CO : IEEE, 1996. P. 336-343. ISBN 0-8186-7508-X.
- 104 *Cyganiak Richard, Reynolds Dave, Tennison Jeni.* The RDF Data Cube Vocabulary [Electronic resource] // W3C. 2014. Mode of access : <http://www.w3.org/TR/vocab-data-cube/> (29.12.2014).
- 105 *Atemezing G A, Troncy R.* Towards a Linked-Data based Visualization Wizard // International Workshop on Consuming Linked Data. Riva del Garda, Italy : s.n., 2014.
- 106 *Tukey John.* Exploratory Data Analysis. Addison-Wesley Publishing Company, 1977. ISBN 9780201076165.
- 107 *Brunetti Josep Maria, Auer Sören, García Roberto.* The linked data visualization model // International Semantic Web Conference. 2012.
- 108 *Klímek Jakub, Helmich Jiří, Nečaský Martin.* Application of the linked data visualization model on real world data from the czech LOD cloud // International Workshop on Linked Data on the Web. Seoul, Korea : s.n., 2014.
- 109 *van Kesteren Anne.* Cross-Origin Resource Sharing [Electronic resource] // W3C. 2014. Mode of access : <http://www.w3.org/TR/cors/> (29.12.2014).
- 110 *Pietschmann Stefan, Voigt Martin, Rümpel Andreas, Meißner Klaus.* CRUISe: Composition of rich user interface services // Lecture Notes in Computer Science. 2009. Vol. 5648. P. 473-476. ISBN 978-3-642-02817-5.
- 111 *Kushalnagar N, Montenegro G, Schumacher C.* IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals [Electronic resource] // The Internet Engineering Task Force. 2007. Mode of access : <https://tools.ietf.org/html/rfc4919> (30.12.2014).

112 *Kuladinithi K, Bergmann O, Pötsch T, Becker M, Görg C.* Implementation of coap and its application in transport logistics // Proceedings of International Workshop IP+SN. Chicago, Illinois : s.n., 2011.

113 *O'Connell Brian, Stanford-Clark Andy, Gale Martin.* Using the IBM Lotus Expeditor micro broker MQTT client to publish messages [Electronic resource] // IBM. 2007. Mode of access : <http://www.ibm.com/developerworks/lotus/library/expeditor-mqtt/> (30.12.2014).

114 *Lane Dale.* Push notifications for mobile apps [Electronic resource] // dale lane blog. 2011. Mode of access : <http://dalelane.co.uk/blog/?p=938> (30.12.2014).

115 *Stanford-Clark Andy, Linh Truong Hong.* MQTT For Sensor Networks (MQTT-SN) Protocol Specification: Version 1.2 [Electronic resource] // MQTT Home. 2013. Mode of access : [http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN\\_spec\\_v1.2.pdf](http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf) (30.12.2014).

116 Mobile Devices announcement [Electronic resource] // Mobile Devices Home. Mode of access : <http://www.mobile-devices.com/c4evo-announcement/> (30.12.2014).

117 PC ECG MONITOR [Electronic resource] // ET COMM Home. Mode of access : [http://www.etcomm.cn/en/products\\_hc-001.html](http://www.etcomm.cn/en/products_hc-001.html) (30.12.2014).

118 *Griffin K, Rosenberg J.* Representational State Transfer (REST) for Feature Configuration in Session Initiation Protocol [Electronic resource] // The Internet Engineering Task Force. 2008. Mode of access : <http://tools.ietf.org/pdf/draft-griffin-bliss-rest-00.pdf> (30.12.2014).

119 *Shelby Z, Hartke K, Bormann C, Frank B.* Constrained Application Protocol (CoAP) [Electronic resource] // The Internet Engineering Task Force. 2012. Mode of access : <http://tools.ietf.org/html/draft-ietf-core-coap-12> (30.12.2014).

120 *Babar Sachin, Mahalle Parikshit, Stango Antonietta, Prasad Neeli, Prasad Ramjee.* Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT) //

Communications in Computer and Information Science. 2010. Vol. 89. P. 420-429. ISBN 978-3-642-14477-6.

121 *Vermesan Ovidiu, Friess Peter, Guillemin Patrick, Gusmeroli Sergio, Sundmaeker Harald, Bassi Alessandro, Soler Jubert Ignacio, Mazura Margaretha, Harrison Mark, Eisenhauer Markus, Doody Pat.* Internet of Things Strategic Research Roadmap [Electronic resource] // European Research Cluster on Internet of Things. 2011. Mode of access : [http://www.internet-of-things-research.eu/pdf/IoT\\_Cluster\\_Strategic\\_Research\\_Agenda\\_2011.pdf](http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf) (30.12.2014).

122 *Polk Tim, Turner Sean.* Security challenges for the internet of things // Workshop on Interconnecting Smart Objects with the Internet. Prague : s.n., 2011.

123 *Cavoukian Ann.* Privacy by Design: The Answer to Overcoming Negative Externalities [Electronic resource] // Information and Privacy Commissioner of Ontario. 2009. Mode of access : <http://www.ipc.on.ca/images/Resources/2009-06-23-TrustEconomics.pdf> (30.12.2014).

124 *Roman Rodrigo, Zhou Jianying, Lopez Javier.* On the features and challenges of security and privacy in distributed internet of things // Computer Networks. 2013. Vol. 57. P. 2266-2279.

125 *Mahalle Parikshit, Babar Sachin, Prasad Neeli R, Prasad Ramjee.* Identity Management Framework towards Internet of Things (IoT): Roadmap and Key Challenges // Communications in Computer and Information Science. 2010. Vol. 89. P. 430-439. ISBN 978-3-642-14477-6.

126 *Ilie-Zudor Elisabeth, Kemény Zsolt, van Blommestein Fred, Monostori László, van der Meulen André.* A survey of applications and requirements of unique identification systems and RFID techniques // Computers in Industry. 2011. Vol. 62, N 3. P. 227-252.

127 *Takalo-Mattila Janne, Kiljander Jussi, Eteläperä Matti, Soininen Juha-Pekka.* Ubiquitous Computing by Utilizing Semantic Interoperability with Item-Level Object

Identification // Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. 2011. Vol. 68. P. 198-209.

128 *Bauge T.* D3.5 – Global and Pluggable Sensor and Actuator Networking Framework [Electronic resource] // SENSEI Project. 2011. Mode of access : <http://www.sensei-project.eu/> (30.12.2014).

129 *Akram H, Hoffmann M.* Supports for Identity Management in Ambient Environments - The Hydra Approach // International Conference on Systems and Networks Communications. Sliema : IEEE, 2008. P. 371-377. ISBN 978-0-7695-3371-1.

130 *Zhou Wei, Meinel C.* Implement role based access control with attribute certificates // International Conference on Advanced Communication Technology. Phoenix Park, Korea : s.n., 2004. P. 536-540. ISBN 89-5519-119-7.

131 *Bai Guangdong, Yan Lin, Gu Liang, Guo Yao, Chen Xiangqun.* Context-aware usage control for web of things // Security and Communication Networks. 2014. Vol. 7, N 12. P. 2696-2712.

132 *Brachmann M, Loong Keoh Sye, Morschon O G, Kumar S S.* End-to-End Transport Security in the IP-Based Internet of Things // International Conference on Computer Communications and Networks. Munich : IEEE, 2012. P. 1-5. ISBN 978-1-4673-1543-2.

133 NANODTLS: Mocana's, standards-based, full featured, Embedded DTLS Client & Server [Electronic resource] // MOCANA. Mode of access : <http://www.mocana.com/iot-security/nanodtls> (30.12.2014).

134 *Aggarwal Charu C, Yu Philip S.* A General Survey of Privacy-Preserving Data Mining Models and Algorithms // Advances in Database Systems. 2008. Vol. 34. P. 11-52. ISBN 978-0-387-70991-8.

135 *Shen Yun, Pearson Siani.* Privacy Enhancing Technologies: A Review. USA : HP Laboratories, 2011.

- 136 *Oleshchuk V.* Internet of things and privacy preserving technologies // International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology. Aalborg : s.n., 2009. P. 336-340. ISBN 978-1-4244-4066-5.
- 137 *Teixeira Thiago, Hachem Sara, Issarny Valérie, Georgantas Nikolaos.* Service Oriented Middleware for the Internet of Things: A Perspective // Lecture Notes in Computer Science. 2011. Vol. 6994. P. 220-229. ISBN 978-3-642-24754-5.
- 138 *Ferber Jacques, Gutknecht Olivier, Michel Fabien.* From Agents to Organizations: An Organizational View of Multi-agent Systems // Lecture Notes in Computer Science. 2004. Vol. 2935. P. 214-230.
- 139 *Aberer Karl, Hauswirth Manfred, Salehi Ali.* A middleware for fast and flexible sensor network deployment // Proceedings of the 32nd international conference on Very large databases. Seoul, Korea : s.n., 2006. P. 1199-1202.
- 140 *Gibbons P B, Karp B, Ke Y, Nath S, Seshan S.* IrisNet: an architecture for a worldwide sensor Web // IEEE Pervasive Computing. 2003. Vol. 2, N 4. P. 22-33.
- 141 *Havlik Denis, Gerald Schimak, Denzer Ralf, Stevenot Bernhard.* Introduction to SANY (sensors anywhere) integrated project // Proceedings of the 20th International Conference on Informatics for Environmental Protection. Graz, Austria : s.n., 2006.
- 142 *Ledlie Jonathan, Shneidman Jeff, Welsh Matt, Roussopoulos Mema, Seltzer Margo.* Open problems in data collection networks // Proceedings of the 11th workshop on ACM SIGOPS European workshop. Leuven, Belgium : ACM, 2004.
- 143 *Shneidman Jeff, Pietzuch Peter, Ledlie Jonathan, Roussopoulos Mema.* Hourglass: An Infrastructure for connecting sensor networks and applications. USA : Harvard, 2004.
- 144 A Functional Ontology of Observation and Measurement // Lecture Notes in Computer Science. 2009. Vol. 5892. P. 26-43. ISBN 978-3-642-10435-0.

145 *Page Kevin, De Roure David, Martinez Kirk, Sadler Jason, Kit Oles.* Linked Sensor Data: RESTfully serving RDF and GML // International Workshop on Semantic Sensor Networks. 2009.

146 *Sequeda Juan, Corcho Oscar.* Linked Stream Data: A Position Paper // International Workshop on Semantic Sensor Networks. 2009.

147 *Bolles Andre, Grawunder Marco, Jacobi Jonas.* Streaming SPARQL - Extending SPARQL to Process Data Streams // Lecture Notes in Computer Science. 2008. Vol. 5021. P. 448-462. ISBN 978-3-540-68233-2.

148 *Calbimonte Jean-Paul, Corcho Oscar, Gray Alasdair.* Enabling Ontology-Based Access to Streaming Data Sources // Lecture Notes in Computer Science. 2010. Vol. 6496. P. 96-111. ISBN 978-3-642-17745-3.

149 *Gray Alasdair, Galpin Ixent, Fernandes Alvaro, Paton Norman, Page Kevin, Sadler Jason, Koubarakis Manolis, Kyzirakos Kostis, Calbimonte Jean-Paul, Corcho Oscar, Garcia Raul, Diaz Victor, Liebana Israel.* SemSorGrid4Env Architecture – Phase I [Electronic resource] // Future Internet Assembly: Real World Internet. 2010. Mode of access : [http://rwi.future-internet.eu/images/7/70/SemSorGrid4Env\\_D1\\_3v1.pdf](http://rwi.future-internet.eu/images/7/70/SemSorGrid4Env_D1_3v1.pdf) (26.12.2014).

150 *Nath Suman, Liu Jie, Zhao Feng.* SensorMap for Wide-Area Sensor Webs // Computer. 2007. Vol. 40, N 7. P. 90-93.

151 *Davis Mills.* Semantic Wave 2008 Report: Industry Roadmap to Web 3.0 & Multibillion Dollar Market Opportunities [Electronic resource] // EUROLIB. 2008. Mode of access :

[http://www.eurolibnet.eu/files/REPOSITORY/20090507165103\\_SemanticWaveReport2008.pdf](http://www.eurolibnet.eu/files/REPOSITORY/20090507165103_SemanticWaveReport2008.pdf) (29.12.2014).

152 New Digital Universe Study Reveals Big Data Gap: Less Than 1% of World's Data is Analyzed; Less Than 20% is Protected [Electronic resource] // EMC Corp. 2012. Mode of access : <http://www.emc.com/about/news/press/2012/20121211-01.htm> (29.12.2014).

- 153 Клеменков П А, Кузнецов С Д. БОЛЬШИЕ ДАННЫЕ: СОВРЕМЕННЫЕ ПОДХОДЫ К ХРАНЕНИЮ И ОБРАБОТКЕ // Труды Института системного программирования РАН. 2012 г.. Т. 23. Р. 143-156.
- 154 Скворцов Н А. Отображение моделей данных NoSQL // Труды 14-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». Переславль-Залесский, Россия : s.n., 2012.
- 155 Zaharia M, Chowdhury M, Franklin M J, Shenker S, Stoica I. Spark: Cluster Computing with Working Sets // Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. 2010.
- 156 Berners-Lee Tim, Hendler James, Lassila Ora. The Semantic Web // Scientific American. 2001. Vol. 284, N 5. P. 28-37.
- 157 Dentler Kathrin, Cornet Ronald, ten Teije Annette, de Keizer Nicolette. Comparison of Reasoners for large Ontologies in the OWL 2 EL Profile // Semantic Web. 2011. Vol. 2, N 2. P. 71-87. ISSN 1570-0844.
- 158 Карпов Л Е. Архитектура распределенных систем программного обеспечения. МАКС Пресс, 2007.
- 159 Bergman Mike. Seven Arguments for Semantic Technologies [Electronic resource] // AI3. 2013. Mode of access : <http://www.mkbergman.com/1626/seven-arguments-for-semantic-technologies/> (30.12.2014).
- 160 Belk M, Germanakos P, Papatheocharous E, Constantinides M, Samaras G. Supporting Adaptive Interactive Systems with Semantic Markups and Human Factors // International Workshop on Semantic and Social Media Adaptation and Personalization. Luxembourg : IEEE, 2012. P. 126-130. ISBN 978-1-4673-4563-7.
- 161 Tetlow Phil, Pan Jeff, Oberle Daniel, Wallace Evan, Uschold Michael, Kendall Elisa. Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and

Software Engineering [Electronic resource] // W3C. 2006. Mode of access : <http://www.w3.org/2001/sw/BestPractices/SE/ODA/> (30.12.2014).

162 *Tallevi-Diotallevi Simone, Kotoulas Spyros, Foschini Luca, Lécué Freddy, Corradi Antonio.* Real-Time Urban Monitoring in Dublin Using Semantic and Stream Technologies // Lecture Notes in Computer Science. 2013. Vol. 8219. P. 178-194. ISBN 978-3-642-41337-7.

163 *Llaves Alejandro, Fernández Javier, Corcho Oscar.* Towards Efficient Processing of RDF Data Streams // International Workshop on Ordering and Reasoning. Riva del Garda, Italy : s.n., 2014.

164 *Mladenić Dunja, Škrjanc Maja, Kenda Klemen, Moraru Alexandra, Bradeško Luka, Fortuna Blaž, Škraba Primož.* Semantic data streams and stream ontologies software [Electronic resource] // ENVISION Project. 2012. Mode of access : [http://www.envision-project.eu/wp-content/uploads/2012/11/D4-5\\_v1-01.pdf](http://www.envision-project.eu/wp-content/uploads/2012/11/D4-5_v1-01.pdf) (30.12.2014).

165 *Wetz Peter, Tuan-Dat Trinh, Ba-Lam Do, Amin Anjomshoaa, Kiesling Elmar, Min Tjoa A.* Towards an Environmental Information System for Semantic Stream Data // Proceedings of the 28th International Conference on Informatics for Environmental Protection. Oldenburg, Germany : s.n., 2014. P. 1-8.

166 *Reddy M P, Prasad B E, Reddy P G, Gupta A.* A methodology for integration of heterogeneous databases // IEEE Transactions on Knowledge and Data Engineering. 1994. Vol. 6, N 6. P. 920-933.

167 *Franklin Michael, Halevy Alon, Maier David.* From Databases to Dataspaces: A New Abstraction for Information Management // SIGMOD Rec.. 2005. Vol. 34, N 4. P. 27-33.

168 *Kent William.* The Breakdown of the Information Model in Multi-database Systems // SIGMOD Rec.. 1991. Vol. 20, N 4. P. 10-15.

169 *Breitbart Yuri, Garcia-Molina Hector, Silberschatz Avi.* Overview of Multidatabase Transaction Management // Proceedings of the 1992 Conference of the Centre for

Advanced Studies on Collaborative Research - Volume 2. Toronto, Canada : IBM Press, 1992. P. 23-56.

170 *Kumar Ajay*. Internet of Things (IOT): Seven enterprise risks to consider [Electronic resource] // TechTarget: Search security. 2014. Mode of access : <http://searchsecurity.techtarget.com/tip/Internet-of-Things-IOT-Seven-enterprise-risks-to-consider> (30.12.2014).

## Приложение А

### Отчет о патентных исследованиях

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,  
МЕХАНИКИ И ОПТИКИ»

УДК 347.771.028

№ госрегистрации 114121750065

Инв. №

«УТВЕРЖДАЮ»  
Научный руководитель,  
Университет ИТМО



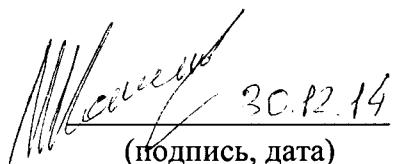
Д.И. Муромцев  
«30» декабря 2014 г.

## ОТЧЕТ О ПАТЕНТНЫХ ИССЛЕДОВАНИЯХ

по теме: «Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)»

Соглашение о предоставлении субсидии от 24 ноября 2014 г. № 14.575.21.0101

Ответственный исполнитель работ:



30.12.14 М.А. Колчин  
(подпись, дата)

Санкт-Петербург 2014

## **Перечень сокращений, условных обозначений, символов, единиц и терминов**

РНТД	– Результаты научно-технической деятельности
ПИ	– Патентные исследования
МПК	– Международная патентная классификация
ПНИ	– Прикладные научные исследования
ПБД	- Патентная база данных
ФИПС	– Федеральный институт промышленной собственности
USPTO	– United State Patent and Trademark Office (Патентное ведомство США)
WIPO	– World Intellectual Property Organization (Всемирная Организация интеллектуальной собственности)
EPO	– European Patent Office (Европейское Патентное ведомство)
JPO	– Japan Patent Office (Японское патентное ведомство)
IOT	- Internet of Things
АП	- Аппаратная платформа
ЦОД	- Центр обработки данных

## **Общие данные об объекте исследований**

В настоящее время наблюдается растущий интерес к использованию технологий распределенных электронных потребительских устройств (Internet of Things), который обусловлен действием следующих факторов:

- а) постоянное удешевление стоимости электронных устройств;
- б) сокращение времени на их производство;
- в) разработка новых протоколов идентификации и передачи данных, в том числе и для электронных устройств с низким энергопотреблением, таких как IPv6, RFID, 6LoWPAN, ZigBee и т.д.;
- г) широкое проникновение информационных технологий практически во все виды производства и жизнедеятельности человека, в том числе, в сельское хозяйство, пищевую и перерабатывающую промышленность, экологический мониторинг, системы безопасности и видеонаблюдения и многие другие.

Примеры уже реализованных проектов, в которых такие технологии нашли свое применение и достаточно эффективно используются, являются:

- а) EkoBus project - проект мониторинга качества воздуха в режиме реального времени в городах Белград и Панчево;
- б) Smart parking project - проект автоматического учета количества свободных мест для парковки в городе.

Исследователи, активно работающие в данной области, выделяют следующие наиболее актуальные научно-технические задачи, связанные с расширением возможностей использования электронных потребительских устройств:

- а) повышение энергоэффективности передачи данных, что позволяет увеличить объем передаваемой информации, хранения данных и уменьшить потребление энергии;

- б) разработка безопасных перепрограммируемых сетей и устройств, в которых одним из наиболее распространенных способов защиты информации является шифрование данных;
- в) расширение количества используемых сервисов за счет применения гетерогенных сетей;
- г) использование протоколов передачи данных, максимально адаптированных к устройствам с низким энергопотреблением;
- д) визуализация данных с использованием GIS-систем для новых типов дисплеев, в том числе 3D-дисплеев;
- е) реализация алгоритмов автоматического межмашинного взаимодействия, обусловленная увеличивающейся интенсивностью обмена информацией и сервисами между распределенными электронными потребительскими устройствами (семантическая интероперабельность);
- ж) извлечение полезной информации (данных) из распределенной сети устройств с различными пространственными и временными разрешающими способностями.

### **A.2.1 Даты начала и окончания работы**

Дата начала работ по проведению патентных исследований: 24.11.2014

Дата окончания работ по проведению патентных исследований: 24.12.2014

### **A.2.2 Краткое описание**

Разработка новых методов и алгоритмов для обеспечения интероперабельности программных и аппаратных компонентов платформы на синтаксическом и семантическом уровнях взаимодействия является одной из актуальных научно-технических проблем в области Internet of Things. Данная проблема, в частности, связана с тем, что компоненты смарт-грид систем крайне разнородны и разнотипны в отношении таких факторов как поставщики оборудования, модели представления данных, протоколы передачи данных и

программные архитектуры. Смарт-грид состоит из множества компонентов на разных уровнях архитектуры («умные счетчики», сенсоры, подстанции и другие), которым необходимо не только обмениваться информацией, но и соответствующим образом контекстуализировать и обрабатывать её («семантическая интероперабельность»). Семантические модели данных описывают конкретную прикладную область с помощью специальных языков, позволяющих проводить вывод на знаниях, таких как языки, основанные на дискреционной логике DAML, OWL, OIL и другие.

Современные подходы к решению проблемы интероперабельности в системах Internet of Things состоят в следующем:

- а) традиционный подход, основанный на разработке специальных программных адаптеров и интерфейсов с использованием традиционных технологий для каждой пары «система – система», для которых необходимо реализовать взаимодействие;
- б) семантический подход, использующий технологии семантического веба, позволяет описывать модели данных и интерфейсы систем, обрабатывать данные в унифицированном виде без применения специальных интерфейсов для каждой пары «система – система».

Традиционный подход позволяет достичь синтаксической интероперабельности через множество специальных программных адаптеров, каждый из которых разрабатывается как отдельный компонент. Его основной недостаток – большие затраты времени на разработку и поддержку таких программ.

В отличие от «семантического» в «традиционном подходе» при разработке гетерогенных систем семантическая интероперабельность как правило не рассматривается. В итоге каждая из систем и/или подсистем является неким «черным ящиком».

Синтаксическая интероперабельность в «семантическом подходе» обеспечивается использованием стандартизированного представления данных, с помощью языка RDF, и семантически связанных данных (Linked Data), а семантическая интероперабельность обеспечивается путем использования «общих словарей» для описания данных (таких как онтологии).

Целью настоящего ПНИ является разработка прототипа масштабируемой децентрализованной программно-аппаратной платформы, использующей:

- а) современные технологии беспроводных сенсорных и агентных сетей (Wireless Sensor and Actor Networks);
- б) технологии семантического веба (Semantic Web technologies);
- в) технологии облачных вычислений для агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things).

Полученные в рамках выполнения настоящих ПНИ результаты планируется использовать в коопeraçãoции с Индустриальным партнером при разработке следующей научно-технической продукции:

- а) умные сети тепло-, водо- и электроснабжения (Smart Grid);
- б) програмная платформа для управления современной городской средой на основе информационно-коммуникационных технологий (Smart City);
- в) универсальная програмная платформа для автоматизированного управления зданиями (Smart Building).

### **A.2.3 Назначение**

Удешевление и сокращение времени разработки и внедрения новой научно-технической продукции при построении умных сетей электроснабжения,

повышение энергоэффективности, надежности, устойчивости производства и распределения электроэнергии.

Обеспечение доступности новых видов информационных услуг за счет применения прогрессивных методов визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных распределенной сети электронных потребительских устройств (Internet of Things), в том числе в режиме реального времени.

Переход от зарубежных аппаратных и программных платформ к отечественным, содействие процессам импорт замещения, переход на более современные и масштабируемые технологии.

#### **A.2.4 Область применения**

Экологический мониторинг, градостроительство, управление инфраструктурой и энергосистемами, промышленные и медицинские системы, системы здравоохранения, уведомления о чрезвычайной ситуации, автоматизация зданий (управление светом, отоплением, вентиляцией, кондиционированием воздуха, системами повышения комфорта, безопасности, энергоэффективности, коммуникационными системами).

### **Основная (аналитическая) часть**

Патентные исследования по теме «Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)» проведены в соответствии с Заданием (Приложение Б) и Регламентом поиска (Приложение В).

В результате поиска выявлено 45 технических решения, имеющих отношение

к объекту исследования (Приложение Г), из которых отобрано 7 релевантных для дальнейшего анализа уровня техники и технического уровня способов агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things).

### **A.3.1 Динамика патентования объекта исследований**

На рисунке А.1 приведена динамика патентования способов мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things), основанная на зависимости количества поданных заявок по годам.

Известно, что дата подачи заявки (дата приоритета) более точно отражает динамику патентования по сравнению с датой выдачи патента. Данное обстоятельство объясняется тем, что фиксированный срок выдачи патента отсутствует. Более того, решение о выдаче патента может быть принято не по каждой заявке. Следует принять во внимание, что данные за последние 1,5-2 года (2012-2014 гг.) являются неполными, так как информация о поданной заявке становится общедоступной, как правило, после ее опубликования, т.е. спустя 18 месяцев с даты ее подачи.

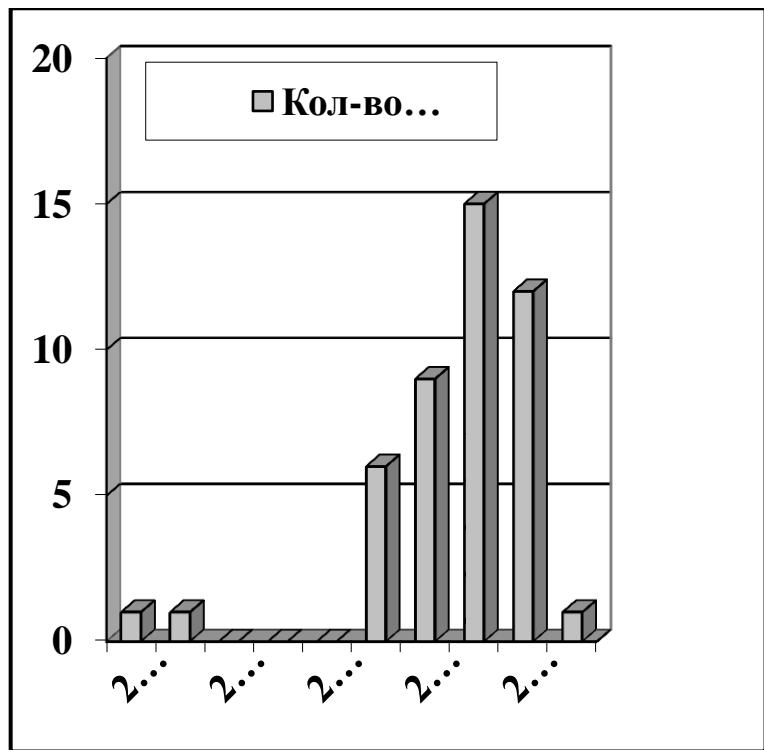


Рисунок А.1 - Динамика подачи заявок на выдачу патентов на способы мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)

### **A.3.2 Анализ способов мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)**

При анализе уровня техники и технического уровня способов агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things), принимались во внимание:

- Технический результат и (или) решаемая задача и (или) цель;
- Стек протоколов передачи данных от электронных потребительских устройств;
- Топология сети электронных потребительских устройств;
- Стек протоколов передачи данных от шлюза к ЦОД;
- Тип аппаратной архитектуры шлюза.

### **A.3.2.1 Исследование уровня техники**

**CN 103415088 (A):** Изобретение относится к распределенным электронным потребительским устройствам (Internet of Things), включает блок электропитания, средство для доступа в Интернет, микропроцессорное устройство, блок UART и модуль для беспроводного доступа. Интеллектуальный коммуникационный шлюз, действующий в гетерогенной сети потребительских устройств, связывается с устройством чтения данных в зашитом режиме, а в случае чрезвычайной ситуации может использовать беспроводную связь. Коммуникационный шлюз обеспечивает диспетчеризацию потока данных, что позволяет значительно сократить размер данных и уменьшить нагрузку на центр обработки данных (далее - ЦОД).

Ограничением и недостатком данного коммуникационного шлюза является необходимость проводного подключения к сети Интернет для передачи данных от электронных потребительских устройств к ЦОД. Так же данное решение ограничивается сбором данных с устройств по проводным протоколам RS232 и RS485, что ограничивает его область применения.

Другим недостатком данного шлюза, является только частичное решение проблемы передачи данных в гетерогенной сети электронных потребительских устройств и обеспечить интероперабельность на большинстве уровнях сетевой модели OSI. Но не на прикладном уровне, где проблема обеспечения интероперабельности заключается в различии схем данных, используемых единиц измерения, их частоты обновления и т.п.

**CN 203167288 (U):** Аппаратная платформа (далее - АП) для разработки коммуникационного шлюза, обеспечивающего доступ к данным распределенной и гетерогенной сети электронных потребительских устройств. АП состоит из материнской платы и подключаемого модуля. АП предоставляет возможность подключение к материнской плате следующих модулей: модуль Bluetooth, модуль ZigBee, модуль 6LoWPAN, модуль RFID, модуль WiFi, модуль 3G/2G. Каждый из подключаемых модулей содержит вычислительное ядро. Используя такой

модульный подход, АП позволяет организовать межмашинное взаимодействие и доступ к данным электронных потребительских устройств различных моделей и поддерживаемых форматов данных по сети Интернет.

Возможность подключения различных коммуникационных модулей к материнской плате является как преимуществом, так и ограничением данного решения. Преимущество заключается в том, что АП позволяет организовать подключение к электронным устройствам, поддерживающим хотя бы один из протоколов передачи данных. Недостатком же является необходимость предоставлять достаточный уровень питания и вычислительных ресурсов для подключаемых модулей с минимальными требованиями не ниже наиболее требовательного модуля.

Такие коммуникационные шлюзы позволяют частично решить проблему передачи данных в гетерогенной сети электронных потребительских устройств и обеспечить интероперабельность на большинстве уровнях сетевой модели OSI. Но на прикладном уровне проблема обеспечения интероперабельности заключается в различии схем данных, используемых единиц измерения, их частоты обновления и т.п.

**CN 202750239 (U):** Известна полезная модель, которая относится к интеллектуальному коммуникационному беспроводному шлюзу медицинского назначения, основанному на технологии распределенных электронных потребительских устройств (Internet of Things). Шлюз содержит микроконтроллер, модуль управления электропитанием и центральный узел беспроводной сенсорной сети. Шлюз характеризуется тем, что микроконтроллер соединен с контролльным приемо-передающим Wi-Fi модулем и коммуникационным управляющим 3G модулем посредством цепи управления выбором сети, Приемо-передающий Wi-Fi модуль соединен с приемо-передающей 2.4G антенной, а коммуникационный управляющий 3G модуль соединен с приемо-передающей 3G антенной. Собранные в шлюзе основные показатели жизнедеятельности могут быть отосланы в облачный

центр обработки данных через Wi-Fi или 3G сеть. Шлюз обеспечивает простую и удобную организацию сети, при этом могут быть удовлетворены разнообразные сетевые требования. Соединение с медицинской техникой может быть осуществлено эффективно и быстро. Может быть реализована гибкая, непрерывная беспроводная сеть. Шлюз потребляет мало энергии, обладает невысокой ценой. Выбор Wi-Fi при совместной работе Wi-Fi и 3G может снизить текущие расходы.

Одним из ограничением данного коммуникационного шлюза является необходимость при отсутствии связи с центром обработки данных (далее - ЦОД) по WiFi в использовании 3G связи, что повышает затраты на сбор данных с медицинского оборудования.

Как и большинство таких коммуникационных шлюзов, они позволяют только частично решить проблему передачи данных в гетерогенной сети электронных потребительских устройств и обеспечить интероперабельность на большинстве уровнях сетевой модели OSI. Но не на прикладном уровне, где проблема обеспечения интероперабельности заключается в различии схем данных, используемых единиц измерения, их частоты обновления и т.п.

**CN 202679655 (U):** Полезная модель представляет многополосный гетерогенный шлюз, созданный по технологии распределенных электронных потребительских устройств (Internet of Things) и относится к области беспроводной передачи данных. Центральный контроллер трансплантируется во внедренную операционную систему Linux, где протоколы многоканального взаимодействия интернета физических объектов трансплантированы в систему, поддерживаются: стандарт IEEE 802.15.4, стандарт IEEE 802.15.4c, стандарт ISO/IEC 18000-4 и стандарт 18000-4. Устройство поддерживает взаимосвязь и межсетевое взаимодействие между электронными потребительскими устройствами и многополосной глобальной коммуникационной сетью посредством IP-связи и мобильной коммуникационной сетью. Полезная модель может одновременно поддерживать доступ к данным высокочувствительной аппаратуры с

многополосными и многоканальными протоколами, обеспечивает также пропускной режим сетевого уровня и позволяет осуществлять прямой обмен данными между многополосным узлом в зоне обнаружения и внешней сетью.

Ограничением данного коммуникационного шлюза является его требования к энергопотреблению и вычислительным ресурсам, так как для обеспечения работы используется операционная система общего назначения.

Такие коммуникационные шлюзы позволяют частично решить проблему передачи данных в гетерогенной сети электронных потребительских устройств и обеспечить интероперабельность на большинстве уровнях сетевой модели OSI. Но прикладном уровне, где проблема обеспечения интероперабельности заключается в различии схем данных, используемых единиц измерения, их частоты обновления и т.п.

**CN 202679656 (U):** Полезная модель относится к сенсорному устройству захвата общего назначения, основанному на коммуникационной технологии ZigBee. Устройство захвата обеспечивает обычный сенсор способностью самостоятельного создания сети и способностью самоуправления, упрощает режим применения сенсора и снижает стоимость эксплуатации. Устройство может быть соединено с сенсорным аппаратом, работающим в диапазоне 0-10mA/0-20mA/4-20mA/0-5V/0-10V и имеющим точку переключения. Устройство может локально осуществлять функцию отображения значений и оптоакустическую сигнальную функцию, строить беспроводную сенсорную сеть через шлюз ZigBee, осуществлять взаимодействие данных с ЦОД и on-line мониторинг в режиме реального времени.

Недостатком этого решения является его высокая стоимость за счет использования модуля ZigBee для передачи данных от электронного потребительского устройства к ЦОД. Стоимость ZigBee модуля превышает стоимость Wi-Fi модуля в 2-4 раза в зависимости от модели передатчика.

Другим недостатком данного шлюза, является только частичное решение проблемы передачи данных в гетерогенной сети электронных потребительских устройств и обеспечить интероперабельность на большинстве уровнях сетевой модели OSI. Но не на прикладном уровне, где проблема обеспечения интероперабельности заключается в различии схем данных, используемых единиц измерения, их частоты обновления и т.п.

**СН 202602709 (U):** Полезная модель относится к технологии распределенных электронных потребительских устройств (Internet of Things) и представляет новый шлюз. Шлюз содержит устройство с внешним интерфейсом, беспроводное устройство, беспроводное поддерживающее устройство модуля доступа и клиентское устройство. Беспроводное устройство соединено с устройством с внешним интерфейсом. Беспроводное поддерживающее устройство модуля доступа соединено с беспроводным устройством. Клиентское устройство соединено с беспроводным поддерживающим устройством модуля доступа. Новый шлюз может сообщаться с каждым узлом беспроводной сенсорной сети ZigBee для одновременного контроля и координации коммуникации между беспроводной сенсорной сетью. Проводная связь может быть реализована посредством нового шлюза и компьютера через интернет-интерфейс. Реализован пакет протоколов базовой программы управления, который поддерживает связь ZigBee. Новый шлюз способен устанавливать динамическую связь с новым узлом в беспроводной сенсорной сети.

Недостатком этого решения является его высокая стоимость за счет использования модуля ZigBee для передачи данных от электронного потребительского устройства к ЦОД. Стоимость ZigBee модуля превышает стоимость Wi-Fi модуля в 2-4 раза в зависимости от модели передатчика.

Другим недостатком данного шлюза, является только частичное решение проблемы передачи данных в гетерогенной сети электронных потребительских устройств и обеспечить интероперабельность на большинстве уровнях сетевой

модели OSI. Но не на прикладном уровне, где проблема обеспечения интероперабельности заключается в различии схем данных, используемых единиц измерения, их частоты обновления и т.п.

**CN 202565475 (U):** Полезная модель относится к технологии распределенных электронных потребительских устройств (Internet of Things) и представляет интеллектуальное мультичастотное интегрированное средство связи, в котором применяется технология беспроводной связи (Wi-Fi) и радиочастотной идентификации (RFID). Устройство может совмещаться со всеми терминалами Wi-Fi и RFID, существующими на рынке. Доступ к сигналам данных электронных потребительских устройств осуществляется в режиме интеграции. Для получения результатов расчетов могут использоваться портативные терминалы. Поддерживается агрегирование данных вычислений в облаке, что позволяет снизить расходы пользователей. Могут одновременно организовываться две и более сети при использовании одного беспроводного устройства и устройства считывания-записи радиочастотной идентификации.

Ограничением данного коммуникационного шлюза является поддержка работы только с RFID-метками, их чтение и запись.

Такие коммуникационные шлюзы не позволяют решить проблему интероперабельности взаимодействия электронных потребительских устройств между собой и с ЦОД на прикладном уровне, где эта проблема заключается в различии схем данных, используемых единиц измерения, их частоты обновления и т.п.

### **A.3.2.2 Исследование технического уровня**

Исследование технического уровня способов мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things) приведено в таблице А.1.

Таблица А.1 - Качественные и количественные характеристики способов мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)

Источник патентной информации, название, приоритет	Технический результат и (или) решаемая задача и (или) цель	Характеристики			
		Стек протоколов передачи данных от электронных потребительских устройств	Топология сети электронных потребительских устройств	Стек протоколов передачи данных от шлюза к ЦОД	Тип аппаратной архитектуры шлюза
CN 103415088 (A) Intelligent heterogeneous Internet-of-Things gateway 03.08.2013	Использование единого совместного коммуникационного шлюза для объединения проводной и беспроводной связи	RS232, RS485, RFID	Звезда	Ethernet	Монолитная
CN 203167288 (U) Internet of Things gateway development platform facing heterogeneous network environments 26.01.2013	Аппаратная платформа для разработки коммуникационного шлюза, обеспечивающего доступ к данным распределенной и гетерогенной сети электронных потребительских устройств	Bluetooth, ZigBee, 6LoWPAN, RFID, WiFi	Звезда, пириングовая сеть	WiFi, 3G/2G, Ethernet	Модульная
CN 202679655 (U) Multi-band heterogeneous internet of things gateway device 24.07.2012	Создание гетерогенного шлюза в сети электронных потребительских устройств, поддерживающего доступ к данным высокочувствительной аппаратуры с многополосными и многоканальными протоколами	WiFi	Звезда	WiFi, 3G, Ethernet	Монолитная

Продолжение таблицы А.1

Источник патентной информации, название, приоритет	Технический результат и (или) решаемая задача и (или) цель	Характеристики			
		Стек протоколов передачи данных от электронных потребительских устройств	Топология сети электронных потребительских устройств	Стек протоколов передачи данных от шлюза к ЦОД	Тип аппаратной архитектуры шлюза
CN 202602709 (U) Novel gateway for Internet of Things 11.04.2012	Создание шлюзов в сети электронных потребительских устройств, способного устанавливать динамическую связь с новым узлом в беспроводной сенсорной сети	RFID, ZigBee	Звезда, пириговая сеть	Ethernet	Монолитная
CN 202565475 (U) Internet of things intelligent multifrequency integration communication device 12.03.2012	Разработка шлюза, имеющего точку беспроводного доступа и обладающего функциями мультичастотной интегрированной связи в сети электронных потребительских устройств и радиочастотной идентификации (RFID)	WiFi, RFID	Звезда	WiFi	Монолитная

### A.3.3 Предложения по использованию результатов патентных исследований

Результаты настоящих патентных исследований совместно с результатами дополнительных патентных исследований могут быть использованы на последующих этапах ПНИ для следующих целей:

- а) актуализации результатов патентных исследований на момент окончания ПНИ.

- б) установления способности к правовой охране результатов интеллектуальной деятельности, полученных при выполнении ПНИ.
- в) анализа потенциальных возможностей применения результатов интеллектуальной деятельности, полученных при выполнении ПНИ.
- г) выбора направлений исследований для создания новых и модернизации существующих объектов техники.
- д) анализа патентной чистоты продукции (объектов техники) в случае ее создания при выполнении ПНИ в отношении конкретных стран, на территории которых предполагается ее использование.

## **Выводы и заключение**

В результате проведенных в настоящем отчете патентных исследований, анализа уровня техники и технического уровня способов мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things) (см. таблицу А.1) установлено следующее:

- а) в период с 2010 года по настоящее время наблюдается рост числа заявок на изобретения, связанных с разработкой новых способов мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things). Их целью является расширение технических возможностей коммуникационных устройств и средств программного обеспечения по созданию условий для эффективного взаимодействия машинного взаимодействия в сети Internet с использованием технологии распределенных электронных потребительских устройств (см. рисунок А.1).
- б) среди используемых стеков протоколов передачи данных от электронных потребительских устройств наиболее востребованными являются WiFi и/или RFID (57 %), которые, как правило, используются в комбинации с другими протоколами, в частности, ZigBee (28 %), а также Bluetooth, 6LoWPAN, RS232, RS485, Serial port (15 %).

- в) самой распространенной топологией сети электронных потребительских устройств является звезда (57 %) или звезда и ее альтернатива – пириングовая сеть (28 %). В оставшихся 15 % случаях сведения о конкретной топологии сети не приводятся.
- г) основными стеками протоколов передачи данных от шлюза к ЦОД являются Ethernet и WiFi (около 60%), которые, как правило, дополняют друг друга или используют в качестве альтернатив такие протоколы как 3G и/или 3G/2G и/или ZigBee.
- д) в подавляющем большинстве проанализированных патентов используется аппаратная архитектура шлюза монолитного тип (85 %) и лишь в 15 % случаев - модульная.

Результаты проведенных патентных исследований рекомендуется учитывать при разработке способов агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things).

## Приложение Б

Задание №1-ИОТ/14 на проведение патентных исследований

**УТВЕРЖДАЮ:**

Научный руководитель,  
Университет ИТМО



Муромцев Д.И.

«30» декабря 2014 г.

**Б.1 Основание для проведения патентных исследований:** соглашение от 24.11.2014 г. № 14.575.21.0101 (п. 3.2, раздел 5, п.6.1.1 технического задания, п. 1.2 этапа 1 плана-графика исполнения обязательств)

**Б.2 Наименование работы (темы) государственного контракта:**  
«Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)»

**Б.3 Шифр:** «2014-14-576-0149-028»

**Б.4 Уникальный идентификатор прикладных научных исследований:**  
«RFMEFI57514X0101»

**Б.5 Этап государственного контракта:** первый

**Б.6 Сроки выполнения:** начало: 24.11.2014 г., окончание: 24.12.2014 г.

**Б.7 Задачи патентных исследований:**

Б.7.1 Анализ уровня техники и технического уровня способов агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для

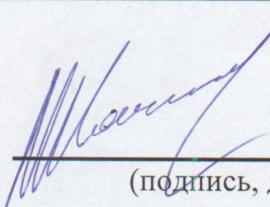
мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)

**Б.8 Ключевые слова:** семантический интернет вещей (semantically enabled internet of things), семантические технологии (semantic technologies), облачные вычисления (cloud computing), онтологии (ontologies), обработка сложных событий (complex event processing), сервис-ориентированная архитектура (service-oriented architecture), беспроводные сенсорные сети (wireless sensor networks), обработка потоковых RDF данных (RDF stream processing), интернет вещей (internet of things), визуализация гетерогенных данных (visualisation of heterogeneous data), интеграция гетерогенных данных (integration of heterogeneous data), анализ потоковых данных (analysis of streaming data), анализ гетерогенных данных (analysis of heterogeneous data), публикация потоковых данных (publishing of streaming data)

#### КАЛЕНДАРНЫЙ ПЛАН

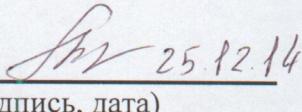
Виды патентных исследований	Подразделения-исполнители (соисполнители)	Ответственные исполнители (Ф.И.О., должность)	Сроки выполнения патентных исследований. Начало. Окончание	Отчетные документы
Анализ технического уровня, уровня техники	Кафедра информатики и прикладной математики	Колчин М.А., инженер	24.11.14 – 24.12.14	Отчет

Ответственный исполнитель

  
25.12.14  
(подпись, дата)

М.А. Колчин

Руководитель патентного подразделения:  
к.т.н., доцент

  
25.12.14  
(подпись, дата)

Л.Н. Казар

## Приложение В

### Регламент поиска № РАТ-1-ИОТ/14

«26» ноября 2014 г.

**В.1 Наименование работы (темы):** «Разработка прототипа масштабируемой сервис-ориентированной программно-аппаратной платформы на основе беспроводных сенсорных и агентных сетей, технологий семантического веба и облачных вычислений в целях агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных в распределенной сети электронных потребительских устройств (Internet of Things)»

**В.2 Шифр:** «2014-14-576-0149-028»

**В.3 Уникальный идентификатор прикладных научных исследований:**  
«RFMEFI57514X0101»

**В.4 Номер и дата утверждения задания:** 1-ИОТ/14 от «24» ноября 2014 г.

**В.5 Этап работы:** первый

**В.6 Цель поиска информации:** исследование технического уровня, анализ уровня техники

В.6.1 Уровень техники фиксирует наилучшие достижения, направленные на решение конкретной прикладной задачи (изобретение), и поэтому имеет определяющее значение для оценки патентоспособности технического решения, относящегося к продукту или способу. Он включает любые сведения, ставшие общедоступными в мире до даты приоритета изобретения. Основной информационной компонентой об уровне техники (не менее 95–99%), является информация, получаемая из патентов, которые призваны обеспечить наиболее широкую и полную правовую охрану самых значимых технико-экономических

параметров разработок, что исключает их использование в реальной продукции без дополнительных исследований и усовершенствований.

**В.6.2 Уровень техники** включает любые сведения, ставшие общедоступными в мире до даты приоритета изобретения. При установлении новизны изобретения в уровень техники также включаются при условии их более раннего приоритета все поданные в Российской Федерации другими лицами заявки на выдачу патентов на изобретения. В настоящее время патентная информация достаточно полно представлена в патентных базах данных национальных, региональных и международных организаций, в том числе доступна в сети Интернет в режиме реального времени. Патентный поиск для актуализации уровня техники проводить по патентным базам данных национальных, региональных и международных организаций в сети Интернет в соответствие с графикой 12 таблицы В.2. «Источники информации, по которым будет проводиться поиск». Наименование патентных источников приведены в таблице В.1.

**В.6.3 Результаты исследования технического уровня, основанного на информации, полученной из уровня техники,** представить в табличном виде в основной (аналитической) части отчета. Для обеспечения аутентичности сведений о технических решениях, приводимых в источниках патентной информации (патентов, заявок на получение патентов), осуществлять их цитирование на языке оригинала, если таким языком является русский или английский язык. Для источников патентной информации, содержащих сведения на иных языках, кроме русского и английского, представить машинный (электронный) перевод соответствующих фрагментов текста на русский или английский язык.

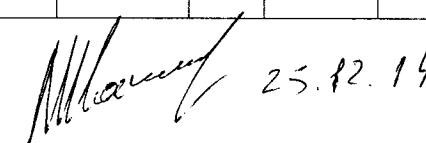
**В.7 Глубина поиска:** не менее 20 лет (использовать документы с 1995 по 2014 гг., а также более ранние документы, если без их учета невозможно оценить уровень техники)

**В.8 Начало:** 24.11.2014 г., **Окончание:** 24.12.2014 г.

Таблица В.1 - Источники информации, по которым будет проводиться поиск

Предмет поиска (объект исследования, его составные части, товар)	Страна поиска	Источники информации, по которым будет проводиться поиск							Ретроспективность	Наименование информационной базы (фонда)
		Наименование	Патентные Классификационные рубрики: МПК (МКИ), МКПО, НКИ и другие	Наименование	НТИ Рубрики УДК и другие	Наименование	Конъюнктурные Код товара: ГС, СМГК, БТН	Наименование		
Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных (полу) структурированных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств	RU  EP US GB DE FR IL JP CN KR SG UA	1. Официальный бюллетень Федеральной службы по интеллектуальной собственности «Изобретения. Полезные модели», М: ФИПС  2. Реферативный журнал «Изобретения стран мира», М: ИНИЦ РОСПАТЕНТА	<b>G06F 09/00</b> G06F 09/44 G06F 17/30 H04L 12/28 H04L 12/58 H04L 12/66 H04L 29/06 H04L 29/08 <b>H04W 04/00</b> H04W 84/18 <b>H04W 88/00</b> H04W 88/10 H04W 88/16	РЖ ВИ- НИ-ТИ	004.04, 004.82, 004.7				1995 – 2014 г. г.	СПбЦНТИ, РНБ, БД в сети Интернет <a href="http://www.pibd.ru">http://www.pibd.ru</a> , <a href="http://www.ptv-vision.ru">http://www.ptv-vision.ru</a> .

Ответственный исполнитель



25.12.14

Колчин М.А.

Руководитель патентного подразделения



25.12.14

Казар Л.Н.

## Приложение Г

### Отчет о поиске

**Г.1 Поиск проведён** в соответствии с заданием научного руководителя Муромцева Д.И. № 1-ИОТ/14 от 24 ноября 2014 г. и регламентом поиска № РАТ-1-ИОТ/14 от 26 ноября 2014 г.

**Г.2 Этап работы:** второй, научно-исследовательская работа

**Г.3 Начало поиска:** 24 ноября 2014 г. **Окончание поиска:** 24 декабря 2014 г.

**Г.4 Сведения о выполнение регламента поиска:** регламент поиска выполнен полностью

**Г.5 Предложения по дальнейшему проведению поиска и патентных исследований:**

Г.5.1 актуализация результатов патентных исследований на момент окончания работ по настоящему государственному контракту;

Г.5.2 установление способности к правовой охране результатов интеллектуальной деятельности, полученных при выполнении настоящего государственного контракта;

Г.5.3 анализ потенциальных возможностей применения результатов интеллектуальной деятельности, полученных при выполнении настоящего государственного контракта;

Г.5.4 выбор направлений исследований для создания новых и модернизации существующих объектов техники;

Г.5.5 анализ патентной чистоты продукции (объектов техники) в случае ее создания при выполнении настоящего государственного контракта в отношении конкретных стран, на территории которых предполагается ее использование.

## Г.6. Материалы, отобранные для последующего анализа

Таблица Г.1 - Патентная документация

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранного документа или причина его аннулирования (только для анализа патентной чистоты)
1	2	3	4	5	6
1.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 103888362 (A)</b> H04L 12/775 H04W 88/10	SHENZHEN KEBONENG TECHNOLOGY CO LTD [CN] CN 20141135942 04.04.2014 25.06.2014	Multi-band wireless sensor network data router and system	<b>CN 103888362 (A)</b> H04L 12/775 H04W 88/10
2.		<b>CN 103731344 (A)</b> H04L 12/66 H04L 29/08	WUHAN GREENET INFORMATION SERVICE CO LTD [CN] CN 20131722842 24.12.2013 16.04.2012	6LOWPAN Internet-of-things gateway based on linux application layer and achieving method	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранных документов. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
3.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 103475707 (A)</b> H04L 29/08	UNIV TSINGHUA [CN] CN 20131407416 09.09.2013 25.12.2013	Universal system for supporting internet of things	
4.		<b>CN 103415088 (A)</b> H04W 88/16	UNIV NORTHEASTERN [CN] CN 20131335688 03.08.2013 27.11.2013	Intelligent heterogeneous Internet-of-Things gateway	
5.		<b>WO 2014182692 (A1)</b> G06F 09/54	CONVIDA WIRELESS LLC [US] US 201361819871P 06.05.2013 13.11.2014	Internet of Things (IOT) adaptation services	
6.		<b>WO 2014186713 (A2)</b>	CONVIDA WIRELESS LLC [US] US 201361823976P 16.05.2013 20.11.2014	Semantic naming model	
7.		<b>WO 2014186733 (A1)</b> H04L 29/08 H04W 04/00	CONVIDA WIRELESS LLC [US] US 201361823988P 16.05.2013 20.11.2014	Systems and methods for enhanced discovery	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
9.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 203167288 (U)</b> H04W 84/18 H04W 88/16	UNIV JIANGNAN [CN] CN 20132145330U 26.03.2013 28.08.2013	Internet of Things gateway development platform facing heterogeneous network environments	
10.		<b>CN 103200081 (A)</b> H04L 12/66 H04W 84/18	UNIV JIANGNAN [CN] CN 20131102548 26.03.2013 10.07.2013	Heterogeneous network environment oriented Internet of things gateway development platform	
11.		<b>US 2014241373 (A1)</b> H04L 12/725	PASAM ROHIT [US] BARTHUR PRADEEP [US] PASAM BHARAT [US] XAPTUM INC [US] US 201414193616 28.02.2014 US 201361770850P 28.02.2013 28.08.2014	Systems, methods, and devices for adaptive communication in a data communication network	
12.		<b>CN 103218220 (A)</b> G06F 09/44	SHANGHAI XIEYU NETWORK TECHNOLOGY CO LTD [CN] CN 2013158569 25.02.2013 24.07.2013	Dynamic pluggable component based-Internet of things middleware system	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
13.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>WO 2014124318 (A1)</b> H04L 29/08	INTERDIGITAL PATENT HOLDINGS [US] US 201361762530P 08.02.2013 14.08.2014	Method and apparatus for incorporating an internet of things (IOT) service interface protocol layer in a node	
14.		<b>CN 103067475 (A)</b> H04L 29/06 H04L 29/08	UNIV BEIJING POSTS & TELECOMM [CM] CN 20121564609 21.12.2012 24.04.2013	Data transmission method and web of things system	
15.		<b>CN 103002597 (A)</b> H04W 84/18	UNIV DALIAN POLYTECHNIC [CN] CN 20121500878 29.11.2012 27.03.2013	Seamless connection multi-gateway wireless sensor network (WSN) structure	
16.		<b>CN 103810528 (A)</b> G06Q 10/00 G06Q 50/26	WUXI JINTIANYANG LASER ELECTRONIC CO LTD [CN] CN 20121444364 08.11.2012 21.05.2014	Internet of Things smart city method and device	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
17.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 103812885 (A)</b> G01D 21/02 G08C 17/02 H04L 29/08 H04W 84/18	WUXI JINTIANYANG LASER ELECTRONIC CO LTD [CN] CN 20121444363 08.11.2012 21.05.2014	Internet of Things energy metering method and device	
18.		<b>CN 103001840 (A)</b> G05B 19/418 H04L 12/28 H04L 29/08	WUXI JINTIANYANG LASER ELECTRONIC CO LTD [CN] CN 20121438482 07.11.2012 27.03.2013	Method and device for internet of things of intelligent home	
19.		<b>CN 202856795 (U)</b> H04L 29/08	JIANGSU INTERNET OF THINGS RES DEV CT [CN] CN 20122548519U 24.10.2012 03.04.2013	Cloud computing infrastructure system for internet of things storage	
20.		<b>CN 202750239 (U)</b> H04W 88/16	SHANDONG COMP SCIENCE CT [CN] CN 20122423733U 24.08.2012 20.02.2013	Wisdom medical wireless gateway based on internet of things technology	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
21.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга	<b>CN 202679655 (U)</b> H04W 88/16	HEFEI BOYAN INTELLIGENT TECHNOLOGY CO LTD [CN] CN 20122375554U 24.07.2012 16.01.2013	Multi-band heterogeneous internet of things gateway device	
22.	и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 202679656 (U)</b> G06F 13/00 G06F 03/00 H04W 92/00	TIANJIN ZIQI ZHENGTAI TECHNOLOGY DEV CO LTD [CN] CN 20122315408U 02.07.2012 16.01.2013	Wireless serial port equipment connecting device applied to Internet of Things	
23.		<b>CN 202634464 (U)</b> H04L 12/28 H04L 29/08	UNIV SOUTH CHINA TECH [CN] CN 20122244931U 29.05.2012 26.12.2012	Realization device of intelligent household unified platform based on UPnP (universal plug-n-play) protocols	
24.		<b>US 2013290305 (A1)</b> G06F 17/30	IBM [US] CN 20121130414 28.04.2012 31.10.2013	Data filtering in the internet of things	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
25.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полустроктурированных и нестроктурированных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 202602709 (U)</b> H04L 29/08 H04L 29/12 H04W 80/02 H04W 80/04 H04W 84/18 H04W 88/16	WUHAN ZHILIN INFORMATION TECHNOLOGY CO LTD [CN] CN 20122149645U 11.04.2012 12.12.2012		
26.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полустроктурированных и нестроктурированных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 202565475 (U)</b> H04W 88/10	XUDONG CHANG [CN] CN 2012289211U 12.03.2012 28.11.2012	Internet of things intelligent multifrequency integration communication device	
27.		<b>CN 102999582 (A)</b> G06F 17/30	UNIV NANJING POSTS & TELECOMM [CN] CN 20121457779 15.11.2012 27.03.2013	Lightweight rule-based WoT (Web of Things) monitoring system	
28.		<b>WO 2013123445 (A1)</b> H04L 29/08 H04W 04/00	INTERDIGITAL PATENT HOLDINGS [US] US 201261600248P 17.02.2012 22.08.2013	Smart internet of things services	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
29.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 102594871 (A)</b> H04L 29/08	SHANDONG CIVIC SOFTWARE ENG CO [CN] SHANDONG CIVIC SOFTWARE BUSINESS MIDDLEWARE CO LTD [CN] CN 20111459162 31.12.2011 18.07.2012	Data packaging method and system, adapter and Internet of things middleware	
30.		<b>CN 102523148 (A)</b> H04L 12/46 H04L 29/06	SUZHOU BAISIMAI INFORMATION CONSULTING CO LTD [CN] CN 20111449477 29.12.2011 27.06.2012	Method for intelligently optimizing Internet of things	
31.		<b>WO 2013089697 (A2)</b> G06F 15/167	EMPIRE TECHNOLOGY DEV LLC [US] LUCIW WILLIAM [US] WO 2011US64844 14.12.2011 20.06.2013	Semantic cache cloud services for connected devices	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
32.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 102291280 (A)</b> H04L 12/26 H04L 29/08	SHENZHEN HEGUANG MEASUREMENT CONTROL TECHNOLOGY [CN] CN 20111239201 19.08.2011 21.12.2011	Intelligent measuring and controlling Internet of things platform based on cloud computing and measuring and controlling method thereof	
33.		<b>CN 202231739 (U)</b> H04L 12/66	UNIV SHANGHAI SCIENCE & TECH [CN] CN 20112285180U 08.08.2011 23.05.2012	Large-scale internet of things gateway system	
34.		<b>CN 102355482 (A)</b> H04L 29/08	SHANDONG CVIC SOFTWARE ENG CO [CN] SHANDONG INFORS SOFTWARE COMMERCIAL MIDDLEWARE CO LTD [CN] CN 20111213328 28.07.2011 15.02.2012	Data transmission method and equipment thereof	
35.		<b>WO 2012095267 (A1)</b> H04L 12/28	ALCATEL LUCENT [FR] FR 20110050178 10.01.2011 19.07.2012	Deployment of services on a set of real objects with automatic matching	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
36.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 102025577 (A)</b> H04L 12/24 H04L 12/28	UNIV XIDIAN [CN] CN 2011102064 06.01.2011 20.04.2011	Network system of Internet of things and data processing method thereof	
37.		<b>EP 2521048 (A1)</b> G06F 17/30	ALCATEL LUCENT [FR] EP 20110305537 06.05.2011 07.11.2012	Method and system for searching on the internet of things	
38.		<b>CN 102111315 (A)</b> H04L 12/28 H04L 29/08 H04W 88/16	DATANG MOBILE COMM EQUIP CO [CN] CN 20101613443 20.12.2010 29.06.2011	Method, system and equipment for fusing Internet of Things and telecommunication network	
39.		<b>CN 102045896 (A)</b> H04W 88/16 H04W 88/18	ZHONGSHAN IKER DIGITAL TECH CO [CN] CN 20101562325 22.11.2010 04.05.2011	Virtual Internet-of-things gateway system capable of realizing multiprotocol and network self-adapting	
40.		<b>CN 102083227 (A)</b> H04W 74/00 H04W 84/12 H04W 84/18	GCI SCIENCE & TECH CO LTD [CN] CN 20101547628 17.11.2010 01.06.2011	Wireless access device and method	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
41.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>CN 102438333 (A)</b> H04L 12/56 H04W 84/18	ZTE CORP [CN] CN 20101503835 29.09.2010 02.05.2012	Universal wireless sensor network system and information processing method thereof	
42.		<b>CN 102355696 (A)</b> H04L 12/56 H04W 84/18	ZTE CORP [CN] CN 20101503835 29.09.2010 02.05.2012	Universal wireless sensor network system and information processing method thereof	
43.		<b>CN 101917465 (A)</b> H04L 29/08 H04W 84/18 H04W 88/16	YONG REN [CN] CHUNXIAO JIANG [CN] NENGQIANG HE [CN] ZHIHUA LI [CN] YANG YU [CN] ZHE YU [CN] CN 20101242450 02.08.2010 15.12.2010	Network architecture of Internet of things based on mobile gateway	

Продолжение таблицы Г.1

№ п/п	Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа. Классификационный индекс	Заявитель (патентообладатель), страна. Номер заявки, дата приоритета, дата публикации	Название изобретения (полной модели, образца)	Сведения о действии охранных документов или причина его аннулирования (только для анализа патентной чистоты)
44.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	<b>US 2008168420 (A1)</b> G06F 09/44	MITRE CORP [US] US 20070000096 07.12.2007 US 20070900312P 09.02.2007 US 20060377459 17.03.2006 US 20060873248P 07.12.2006 10.07.2008	Semantic system for integrating software components	
45.		<b>US 2007168479 (A1)</b> G06F 15/173	AMERICAN EXPRESS TRAVEL RELATE [US] US 20050275391 29.12.2005 19.07.2007	Semantic interface for publishing a web service to and discovering a web service from a web service registry	

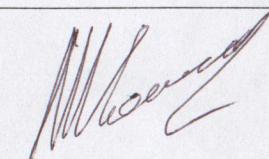
Таблица Г.2 - Научно-техническая, конъюнктурная, нормативная документация и материалы государственной регистрации (отчеты о научно-исследовательских работах)

№ п/п	Предмет поиска	Наименование источника информации с указанием страницы источника	Автор, фирма (держатель) технической документации	Год, место и орган издания (утверждения, депонирования источника)
1	2	3	4	5
1.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурированных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	Internet of Things in Industries: A Survey, IEEE Transactions on Industrial Informatics, 6.874, URL: <a href="http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6714496">http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6714496</a>	Xu, L.	IEEE Transactions on Industrial Informatics. – 2014, DOI: 10.1109/TII.2014.2300753.
2.		Internet of Things (IoT): A vision, architectural elements, and future directions, pp. 1645-1660, DOI: 10.1016/j.future.2013.01.010, URL: <a href="http://www.sciencedirect.com/science/article/pii/S0167739X13000241">http://www.sciencedirect.com/science/article/pii/S0167739X13000241</a>	Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswami	Future Generation Computer Systems, vol. 29, issue 7, 2013
3.		Enabling smart cities through a cognitive management framework for the internet of things, URL: <a href="http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6525602">http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6525602</a> DOI: 10.1109/MCOM.2013.6525602	Vlacheas, P.	IEEE Communications Magazine, 4.46 - 2013
4.		Internet of things: applications and challenges in technology and standardization, pp. 49-69, 2011	D. Bandyopadhyay, and J. Sen	Wireless Personal Communications, vol. 58, no. 1

Продолжение таблицы Г.2

№ п/п	Предмет поиска	Наименование источника информации с указанием страницы источника	Автор, фирма (держатель) технической документации	Год, место и орган издания (утверждения, депонирования источника)
5.	Способы агрегации, нормализации, анализа и визуализации больших массивов гетерогенных структурированных, полуструктурных и неструктурных данных для мониторинга и управления безопасностью распределенной сети электронных потребительских устройств (Internet of Things)	Computational intelligence in wireless sensor networks: a survey, pp. 68–96	R.V. Kulkarni, A. Förster, G.K. Venayagamoorthy	IEEE Communications Surveys & Tutorials 13 (2011)
6.		Assessing Smart Grid Benefits and Impacts: EU and U.S. Initiatives, URL: <a href="http://ses.jrc.ec.europa.eu/sites/ses.jrc.ec.europa.eu/files/documents/eu-us_smart_grid_assessment_-_final_report_-online_version.pdf">http://ses.jrc.ec.europa.eu/sites/ses.jrc.ec.europa.eu/files/documents/eu-us_smart_grid_assessment_-_final_report_-online_version.pdf</a>		
7.		IEC, IEC Smart Grid Standardization Roadmap, URL: <a href="http://www.iec.ch/ smartgrid/downloads/sg3_roadmap.pdf">www.iec.ch/ smartgrid/downloads/sg3_roadmap.pdf</a>		
8.		The Internet of Things: A survey, URL: <a href="http://www.sciencedirect.com/science/article/pii/S1389128610001568">http://www.sciencedirect.com/science/article/pii/S1389128610001568</a>	Luigi Atzori	Computer Networks, 1.871. – 2010, DOI: 10.1016/j.comnet. 2010.05.010
9.		The Internet of things, URL: <a href="http://dx.doi.org/10.1038/scientificamerican1004-76">http://dx.doi.org/10.1038/scientificamerican1004-76</a>	Gershenfeld Neil	Scientific American, 1.328. – 2004, DOI: 10.1038/scientificamerican 1004-76
10.		The Internet of things, URL: <a href="http://dx.doi.org/10.1038/scientificamerican1004-76">http://dx.doi.org/10.1038/scientificamerican1004-76</a>	Gershenfeld Neil	Scientific American, 1.328. - 2004, DOI: 10.1038/scientificamerican 1004-76

Исполнитель отчета



25.12.14

Колчин М.А.

Приложение Д  
Руководство по эксплуатации ПАС

ПРОГРАММНО-АППАРАТНЫЙ СТЕНД

ПАС.000000.001 00 01

## АННОТАЦИЯ

Настоящий документ предназначен для операторов программно-аппаратного стенда для проведения экспериментальных исследований ЭО ПАП.

В документе содержатся сведения о назначении программно-аппаратного стенда и условиях, необходимых для его выполнения. Приведены последовательность действий оператора, обеспечивающие запуск и решение задач, и тексты сообщений выдаваемых в ходе работы.

## **Д.1 НАЗНАЧЕНИЕ ПРОГРАММНО-АППАРАТНОГО СТЕНДА**

Программно-аппаратный стенд предназначен для проведения экспериментальных исследований экспериментального образца масштабируемой сервис-ориентированной программно-аппаратной платформы для сбора, нормализации, обработки и визуализации больших массивов гетерогенных данных в распределенной сети электронных потребительских устройств.

ПАС выполняет следующие функции:

- a) симулирует показания электронных потребительских устройств, таких как общедомовой счетчик тепловой энергии и уличный датчик температуры
- б) предоставляет доступ внешним системам к текущим моделируемым показаниям через программный интерфейс

## **Д.2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММНОГО КОМПЛЕКСА ПАС**

Программный комплекс ПАС (далее ПК ПАС) выполняется в среде операционной среды Ubuntu Server. ПК ПАС требует наличия следующего общесистемного программного обеспечения:

- а) Ubuntu Server 14.10 x64 или выше
- б) Java Virtual Machine 1.8 или выше
- в) MySQL 5.6 или выше

Пользователю, под именем которого осуществляется работа с ПК, должен быть открыт доступ к диску файлового хранилища на чтение и запись.

## **Д.3 ИНСТАЛЛЯЦИЯ ПРОГРАММНОЙ ЧАСТИ ПАС**

### **Д.3.1 Состав дистрибутива ПЧ ПАС**

В состав ПК ПАС входят следующие компоненты:

- а) подсистема симуляции в виде программы simulator.jar
- б) подсистема интеграции данных в виде программы soap-server.jar

Перед инсталляцией программной части ПАС необходимо с помощью подсистемы виртуализации серверов создать виртуальную машину и выполнить установку операционной системы Ubuntu Server 14.10 или выше. А также выполнить установку необходимого программного обеспечения перечисленного в разделе Д.2.

### **Д.3.2 Установка файлов программной части ПАС**

В домашней директории пользователя, под именем которого будет выполняться ПК ПАС, создать директорию phs, в которой разместить файлы программных компонентов, такие как simulator.jar и soap-server.jar. Далее необходимо создать файл конфигурации под названием phs.config, в котором указать необходимые настройки перечисленные в разделе Д.3.3.

Для проверки наличия установленной Java Virtual Machine необходимо выполнить в командном терминале следующую команду: java -version. В случае успешного выполнения в командном терминале должна быть выведен текст как на рисунке Д.1.

```
openjdk version "1.8.0_31"
OpenJDK Runtime Environment (build 1.8.0_31-b13)
OpenJDK 64-Bit Server VM (build 25.31-b07, mixed mode)
```

Рисунок Д.1 – Результат выполнения команды java -version

### **Д.3.3 Настройка программной части ПАС**

Для настройки работы ПК ПАС использует файл конфигурации phs.config установленный в директории phs.

Далее приводится список конфигурационных настроек для подключения к подсистеме хранения:

- а) hibernate.connection.url – указывает URI базы данных, к которой должно быть осуществлено подключение, например, jdbc:mysql://localhost/phs-db
- б) hibernate.connection.username – имя пользователя для подключения
- в) hibernate.connection.password – пароль пользователя для подключения

Данные конфигурационные настройки используются как подсистемой симуляции, так и подсистемой интеграции данных.

Для конфигурирования поведения подсистемы симуляции электронных потребительских устройств используются следующие конфигурационные настройки:

- а) simulation.thermalmeter.amount – устанавливает количество общедомовых счетчиков тепловой энергии для симуляции их показаний
- б) simulation.thermalmeter.interval – интервал между показаниями датчика в минутах
- в) simulation.airtemp.amount – устанавливает количество уличных датчиков температуры для симуляции их показаний
- г) simulation.airtemp.interval – интервал между показаниями датчика в минутах
- д) simulation.month – название месяца года, среднесуточные показания которого будут использоваться для симуляции

## **Д.4 ВЫПОЛНЕНИЕ ПРОГРАММНОГО КОМПЛЕКСА ПАС**

### **Д.4.1 Запуск ПК ПАС**

Запуск программной части программно-аппаратного стенда выполняется в следующей последовательности:

- a) запуск подсистемы симуляции через командный терминал по средствам команды: `java -jar simulator.jar &> simulator.log &`
- b) запуск подсистемы интеграции через командный терминал по средствам команды: `java -jar soap-server.jar &> soap-server.log &`

После выполнения данных команд ПАС готов к обработки запросов по протоколу СоAP на порту 4545. Обе программы выполняются в фоновом режиме, а лог их выполнения доступ в файлах `simulator.log` и `soap-server.log` соответственно.

### **Д.4.2 Остановка ПЧ ПАС**

Для остановки ПК ПАС необходимо убить процессы запущенных программ, а для этого необходимо выполнить следующие действия:

- a) выполнить команду: `ps | grep <имя .jar пакета>`, где `<имя .jar пакета>` должно быть указано `simulator.jar` или `soap-server.jar`
- б) найти в первом столбце вывода последней команды номер процесса
- в) выполнить команду: `kill <номер процесса>`

## **Д.5 СООБЩЕНИЯ ОПЕРАТОРУ**

Во время выполнения приведенных в предыдущих разделах команд оператор может столкнуться с некоторым сообщениями, в соответствии с таблицей Д.2.

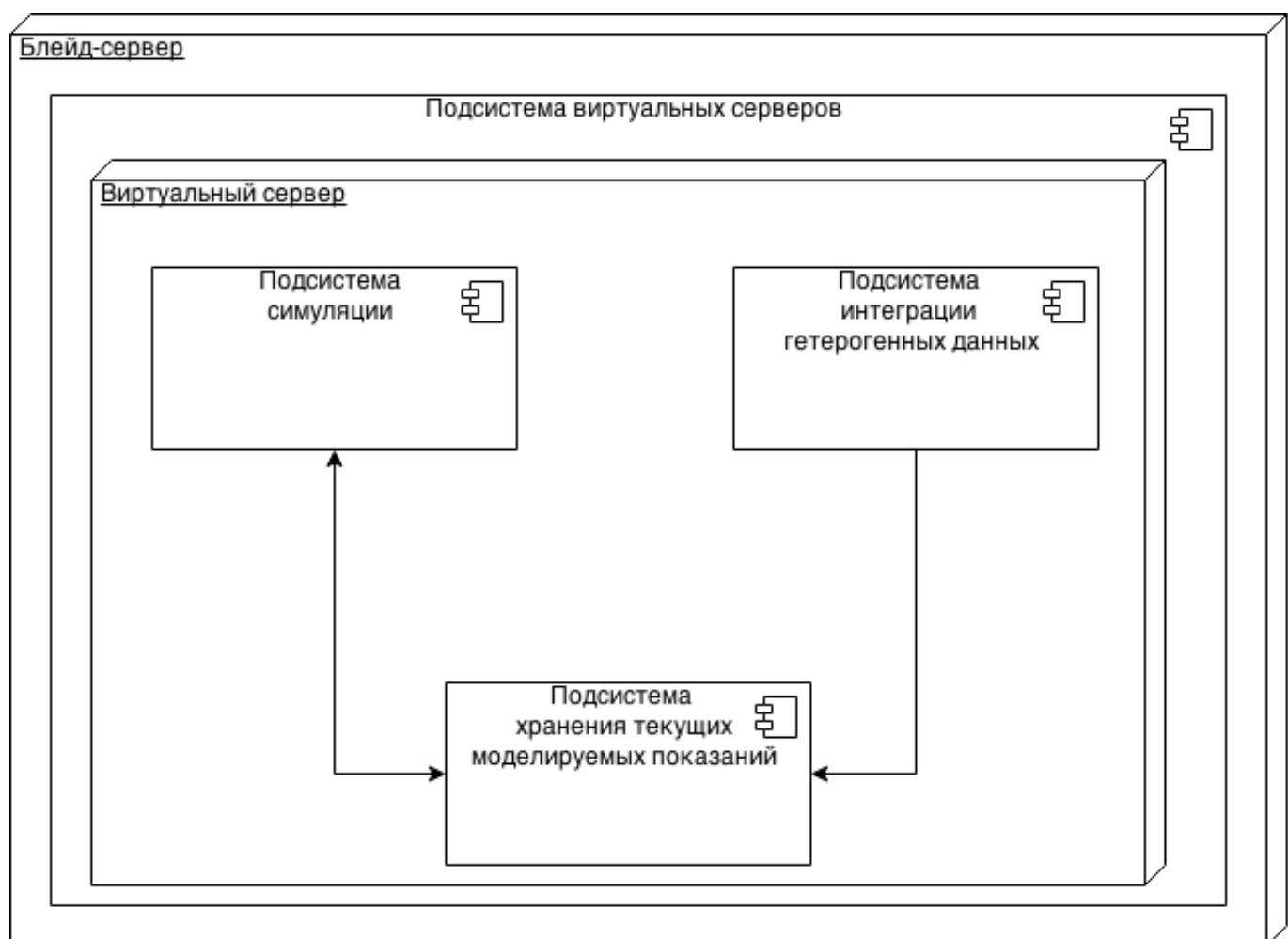
Таблица Д.2 – Таблица с сообщениями и необходимыми действиями

Сообщение	Ситуация
GenericJDBCException: Cannot open connection	В данном случае необходимо: - проверить наличие необходимых настроек подключения к БД - проверить запущена ли БД
Exception: Port 4545 already in use	В данном случае необходимо удостовериться, что ни один из запущенных процессов в системе не использует порт 4545, после чего попробовать запустить программу снова
FileNotFoundException: phs.config (No such file or directory)	В данном случае надо удостовериться о наличии файла конфигурации phs.config в папке с simulator.jar и soap-server.jar
UnsupportedClassVersionError	В данном случае необходимо удостовериться, что версия Java Virtual Machine соответствует требуемой

Лист регистрации изменений								
Изм.	Номера листов (страниц)			Всего листов в в документе	№ документа	Входящий №	Подпись (фамилия)	Дата
	измененных	замененных	новых	аннулированных				

## Приложение Е

### Структурная схема ПАС



Приложение Ж

Формуляр

**503200**  
**(код продукции)**

Программно-аппаратный стенд  
**наименование и индекс изделия**

**ФОРМУЛЯР**  
ПАС.000000.001 ФО 01  
**обозначение документа**

## **Ж.1 Основные сведения об изделии**

Наименование: Программно-аппаратный стенд (ПАС) с подсистемой виртуальных серверов, подсистемой интеграции больших массивов гетерогенных данных и симуляцией распределенных вычислений для моделирования сети электронных потребительских устройств (Internet of Things).

Обозначение: ПАС.000000.001

Дата изготовления: 20.12.2014

Наименование и/или почтовый адрес изготовителя: федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики», 197101, г. Санкт-Петербург, Кронверкский пр., д.49

## **Ж.2 Основные технические данные**

<b>Наименование параметра</b>	<b>Значение</b>
Мощность, Вт	200
Наработка с начала эксплуатации	24 ч.
ОЗУ	12 Гб
Процессор	Intel Xeon 5600/5500
Дисковое пространство	1 Тб
Количество симулируемых электронных потребительских устройств	до 2000

## **Ж.4 Ресурсы, сроки службы и хранения, гарантии изготовителя**

**Ресурс изделия до первого \_\_\_\_\_**  
среднего, капитального

**ремонта \_\_\_\_\_**  
параметр, характеризующий наработку

**в течение срока службы \_\_\_\_\_ лет, в том числе срок хранения \_\_\_\_\_ лет (года)**  
**в консервации (упаковке) изготовителя,**

**в складских помещениях, на открытых площадках и т.п.**

**Межремонтный ресурс \_\_\_\_\_**  
параметр, характеризующий наработку  
**при \_\_\_\_\_ ремонте (ах) в течение срока службы \_\_\_\_\_ лет.**

**Указанные ресурсы, сроки службы и хранения действительны при соблюдении потребителем требований действующей эксплуатационной документации.**

-----  
Линия отреза при поставке на экспорт

**Гарантии изготовителя:** Предприятие-изготовитель гарантирует стабильную работу программно-аппаратного стенда. А так же гарантируется работоспособность элементов программно-аппаратного стенда в соответствии с гарантиями производителей данных элементов.

Предприятие-изготовитель обязуется принимать меры по устранению ошибок в программном-аппаратном стенде при обнаружении в процессе работы и своевременном сообщении о них со стороны пользователей с предоставлением полной информации об условиях возникновения ошибки.

## Ж.5 Консервация

## **Ж.6 Свидетельство об упаковывании**

Программно-аппаратный стенд    ПАС.000000.001    №  
наименование изделия                        обозначение                        заводской номер

**Упакован(а)**\_\_\_\_\_

**наименование или код изготовителя**

**согласно требованиям, предусмотренным в действующей технической документации.**

\_\_\_\_\_  
**должность**

\_\_\_\_\_  
**личная подпись**

\_\_\_\_\_  
**расшифровка подписи**

## **Ж.7 Свидетельство о приемке**

<u>Программно-аппаратный стенд</u>	<u>ПАС.000000.001</u>	
наименование изделия	обозначение	заводской номер

**изготовлен(а) и принят(а) в соответствии с обязательными требованиями государственных (национальных) стандартов, действующей технической документацией и признан(а) годным(ой) для эксплуатации.**

## Ж.8 Движение изделия при эксплуатации

Дата установки	Где установлено	Дата снятия	Наработка		Причина снятия	Подпись лица, проводившего установку (снятие)
			с начала эксплуатации	после последнего ремонта		

### **Ж.8.1 Прием и передача изделия**

Дата	Состояние изделия	Основание (наименование, номер и дата документа)	Предприятие, должность и подпись		Примечание
			сдавшего	принявшего	

## Ж.8.2 Сведения о закреплении изделия при эксплуатации

Наименование изделия (составной части) и обозначение	Должность, фамилия и инициалы	Основание (наименование, номер и дата документа)		Примечание
		Закрепление	Открепление	

## Ж.9 Учет работы изделия

Дата	Цель работы	Время		Продолжительность работы	Наработка	Кто проводит работу	Должность, фамилия и подпись ведущего формулляр
		начала работы	окончания работы				

## **Ж.10 Учет технического обслуживания**

## **Ж.11 Учет работы по бюллетеням и указаниям**

Номер бюллетеня (указания)	Краткое содержание работы	Установленный срок выполнения	Дата выполнения	Должность, фамилия и подпись	
				выполнившего работу	проверившего работу

## **Ж.12 Работы при эксплуатации**

### **Ж.12.1 Учет выполнения работы**

Дата	Наименование работы и причина ее выполнения	Должность, фамилия и подпись		Примечание
		выполнившего работу	проверившего работу	

**Ж.12.2 Периодический контроль основных эксплуатационных и технических характеристик**

## Ж.12.4 Проверка средств измерений

Наименование и обозначение средств измерения	Заводской номер	Дата изготовления	Периодичность поверки	Проверка					Примечание
				Дата	Срок очередной поверки	Дата	Срок очередной поверки	Дата	

## Ж.12.5 Техническое освидетельствование контрольными органами

Наименование и обозначение составной части изделия	Заводской номер	Дата изготовления	Периодичность освидетельствования	Освидетельствование					Примечание
				Дата	Срок очередного освидетельствования	Дата	Срок очередного освидетельствования	Дата	

## Ж.13 Хранение

Дата		Условия хранения	Вид хранения	Примечание
приемки на хранение	снятия с хранения			

## **Ж.14 Ремонт**

### **Ж.14.1 Краткие записи о произведенном ремонте**

наименование изделия \_\_\_\_\_ № \_\_\_\_\_ обозначение \_\_\_\_\_ заводской номер \_\_\_\_\_

Наработка с начала эксплуатации \_\_\_\_\_

параметр, характеризующий ресурс или срок службы

Наработка после последнего  
ремонта \_\_\_\_\_

параметр, характеризующий ресурс или срок службы

Причина поступления в ремонт \_\_\_\_\_

Сведения о произведенном ремонте \_\_\_\_\_

вид ремонта и краткие

сведения о ремонте

## **Ж.14.2 Данные приемосдаточных испытаний**

ПАС соответствует заданным характеристикам.

### **Ж.14.3 Свидетельство о приемке и гарантии**

наименование изделия № \_\_\_\_\_ обозначение \_\_\_\_\_ заводской номер \_\_\_\_\_

вид ремонта \_\_\_\_\_ согласно \_\_\_\_\_  
наименование предприятия, вид документа  
условное обозначение

**Принят(а) в соответствии с обязательными требованиями государственных (национальных) стандартов и действующей технической документацией и признан(а) годным(ой) для эксплуатации.**

**Ресурс до очередного ремонта** \_\_\_\_\_

в течение срока службы \_\_\_\_\_ лет

**(года), в том числе срок хранения** \_\_\_\_\_

условия хранения лет (года).

**Исполнитель ремонта гарантирует соответствие изделия требованиям действующей технической документации при соблюдении потребителем требований действующей эксплуатационной документации.**

## Ж.15 Особые отметки

## Ж.17 Контроль состояния изделия и ведения формуляра

Дата	Вид контроля	Должность проверяющего	Заключение и оценка проверяющего		Подпись проверяющего	Отметка об устранении замечания и подпись
			по состоянию изделия	по ведению формуляра		

Приложение И  
Описание программного обеспечения ПАС

УТВЕРЖДАЮ

Руководитель ПНИ

 Д.И. Муромцев  
«30» декабря 2014 г.

ПРОГРАММНЫЙ КОМПЛЕКС ПРОГРАММНО-АППАРАТНОГО СТЕНДА  
Описание программы

ЛИСТ УТВЕРЖДЕНИЯ

ПАС.000000.001 00 01-ЛУ

Ответственный исполнитель  
 М.А. Колчин  
«30» декабря 2014 г.

УТВЕРЖДЕН

ПАС.000000.001 00 01-ЛУ

ПРОГРАММНЫЙ КОМПЛЕКС ПРОГРАММНО-АППАРАТНОГО СТЕНДА

Описание программы

ПАС.000000.001 00 01

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

## АННОТАЦИЯ

Настоящий документ предназначен для использования в процессе сопровождения программной части программно-аппаратного стенда.

В документе приводятся общие сведения о программного обеспечения ПАС, реализуемые функции, описание логической структуры программного обеспечения, входные и выходные данные, используемые программным обеспечением.

## **И.1 Общие сведения**

### **И.1.1 Обозначение и наименование программного комплекса**

Наименование программного комплекса: программная часть программно-аппаратного стенда предназначенного для проведения экспериментальных исследований экспериментального образца масштабируемой сервис-ориентированной программно-аппаратной платформы для сбора, нормализации, обработки и визуализации больших массивов гетерогенных данных в распределенной сети электронных потребительских устройств.

Обозначение: ПАС.00001-01

### **И.1.2 Окружение программного комплекса**

Программный комплекс требует наличие следующего программных средств:

- а) операционная система Ubuntu Server 14.10 или выше
- б) Java Virtual Machine 1.8 или выше
- в) MySQL 5.6 или выше

### **И.1.3 Система программирования**

Программный комплекс реализован на языке программирования Java.

Использованы следующие программные библиотеки:

- а) MaDKit – библиотека для разработки много-агентных систем и систем симуляции на основе агентного моделирования
- б) Hibernate – библиотека для работы с базами данных и автоматического отображения реляционных таблиц в соответствующие экземпляры классов на языке программирования Java
- в) Californium – библиотека для разработки серверов и клиентов поддерживающих протокол передачи данных СоАР
- г) Owner – библиотека для работы с файлами конфигурации

## **И.2 Функциональное назначение**

### **И.2.1 Назначение программного комплекса**

Программно-аппаратный стенд предназначен для проведения экспериментальных исследований экспериментального образца масштабируемой сервис-ориентированной программно-аппаратной платформы для сбора, нормализации, обработки и визуализации больших массивов гетерогенных данных в распределенной сети электронных потребительских устройств.

### **И.2.2 Решаемые задачи**

Программный комплекс предназначен для решения следующих задач:

- a) симулирует показания электронных потребительских устройств, таких как общедомовой счетчик тепловой энергии и уличный датчик температуры
- б) предоставляет доступ внешним системам к текущим моделируемым показаниям через программный интерфейс

## **И.3 Описание логической структуры**

### **И.3.1 Алгоритм работы программного комплекса**

#### **И.3.1.1 Этапы алгоритма работы**

Программный комплекс симулирует показания электронных потребительских устройств для проведения экспериментальных исследований ЭО ПАС.

Работа комплекса состоит из следующих этапов:

- а) чтение файла конфигурации
- б) подключение к БД и получение данных о среднесуточных показаниях для моделируемых устройств
- в) симуляция показаний устройств в соответствии с настройками установленными в файле конфигурации
- г) запись моделируемых показаний в БД
- д) ожидание и обработка запросов на получение текущих моделируемых показаний устройств

#### **И.3.1.2 Чтение файла конфигурации**

Файл конфигурации располагается в той же директории, что и остальные части комплекса, и называется phs.config. Чтение файла производиться с помощью библиотеки Owner, а формат файла соответствует формату Java Properties.

#### **И.3.1.3 Подключение к базу данных и получение данных**

Подключение к базе данных происходит в соответствии с установленными настройками в файле конфигурации и с использованием библиотеки Hibernate. Получение данных производиться по средствам объектно-ориентированного языка запросов, который транслируется в язык запросов SQL, а в результате запроса полученные данные конвертируются в экземпляры соответствующих классов.

#### **И.3.1.4 Симуляция показаний электронных потребительских устройств**

Симуляция показаний электронных потребительских устройств производится с применением подхода агентного моделирования. Программный комплекс реализует алгоритмы симуляции показаний следующих устройств:

- а) общедомовой счетчик тепловой энергии
- б) уличный датчик температуры

Алгоритмы, описывающие жизненные циклы агентов приводятся в разделе 3.4.3.5 и 3.4.3.6.

#### **И.3.1.5 Запись моделируемых показаний в БД**

Текущие смоделированные показания записываются в базу данных по средствам объектно-ориентированного языка запросов в режиме пакетной обработки, т.е. новые записи записываются в БД не моментально, а происходит их накапливание и запись пакетами.

#### **И.3.1.6 Ожидание и обработка запросов**

Ожидание и обработка запросов на получение текущих моделируемых показаний производится с использование протокола передачи данных СоАР. Для обработки запросов используется реализация этого протокола в виде библиотеки Californium, которая позволяет обрабатывать запросы с помощью функций обратного вызова.

### **И.3.2 Используемые методы**

Методы, используемые программным комплексом, указаны при описании алгоритмов в разделе И.3.1.

### **И.3.3 Структура программного комплекса**

Программный комплекс состоит из следующих программных компонентов:

- а) подсистема симуляции в виде файла simulator.jar
- б) подсистема интеграции данных в виде файла soap-server.jar

в) подсистема хранения данных (база данных) в виде СУБД MySQL 5.6  
или выше

База данных используется другими модулями для хранения и доступа к текущим моделируемым показаниями электронных потребительских устройств. Архитектура программного комплекса приведена в разделе 3.4.1.

## **И.4 Используемые технические средства**

Минимальные требования к аппаратному обеспечению ПАС для обеспечения корректной работы программного комплекса:

- а) оперативная память не менее 2 GByte/CPU
- б) дисковое пространство не менее 20 GByte/CPU
- в) модель процессора не хуже Intel Xeon 5600/5500

## **И.5 Вызов и загрузка**

### **И.5.1 Условия установки программного комплекса**

Программный комплекс ПАС (далее ПК ПАС) выполняется в среде операционной среды Ubuntu Server. ПЧ ПАС требует наличия следующего общесистемного программного обеспечения:

- а) Ubuntu Server 14.10 x64 или выше
- б) Java Virtual Machine 1.8 или выше
- в) MySQL 5.6 или выше

Пользователю, под именем которого осуществляется работа с ПК, должен быть открыт доступ к диску файлового хранилища на чтение и запись.

### **И.5.2 Установка и запуск файлов программного комплекса**

В домашней директории пользователя, под именем которого будет выполняться ПК ПАС, создать директорию phs, в которой разместить файлы программных компонентов, такие как simulator.jar и soap-server.jar. Далее необходимо создать файл конфигурации под названием phs.config, в котором указать необходимые настройки перечисленные в разделе Д.3.3.

Запуск и остановка программного комплекса описана в разделе Д.4.

## **И.6 Входные данные**

### **И.6.1 Состав входных данных**

Входными данными программного комплекса являются:

- а) файл конфигурации программного комплекса, задающий параметры соединения с БД и параметры симуляции
- б) результаты запросов к БД

### **И.6.2 Файл конфигурации программного комплекса**

Настройка программного комплекса осуществляется через редактирование файла конфигурации расположенного в папке phs вместе с остальными файлами комплекса. Файл представляет собой файл в формате Java Properties, где параметры задаются через строки вида [название параметра]=[значение параметра].

Файл конфигурации включает в себя параметры соединения с базой данных, а также параметры задающие поведение подсистемы симуляции, т.е. количество и типы электронных потребительских устройств, показания которых симулируются.

Описание конкретных параметров приводится в разделе Д.3.3 инструкции по эксплуатации.

### **И.6.3 Данные, получаемые из БД**

Схема базы данных программного комплекса приведена в разделе 3.4.4.4. База данных содержит следующие данные необходимые для работы комплекса:

- а) типы электронных потребительских устройств, показания которых моделируются: общедомовой счетчик тепловой энергии и уличный датчик температуры
- а) типы показаний, например, объем потребленной тепловой энергии или температура воздуха
- б) среднесуточные показания для каждого месяца года, которые необходимы для симуляции показаний

## **И.7 Выходные данные**

Выходными данными программного комплекса являются моделируемые показания электронных потребительских устройств. Выходные данные записываются в базу данных и отправляются в ответ на запросы клиентов по протоколу СоАР.

Выходные данные хранятся в базе данных в таблице Показания, которая имеет следующие поля:

- а) Идентификатор показания
- б) Идентификатор сенсор
- в) Идентификатор типа показания
- г) Значение показания
- д) Дата и время записи показания

**Лист регистрации изменений**

И з м .	Номера листов (страниц)				Всего листо в в докум енте	№ документа	Входящи й №	Подпись (фамилия)	Дата
	изме ненн ых	заме ненн ых	нов ых	анну лиро ванн ых					

Приложение К  
Текст программного обеспечения ПАС

УТВЕРЖДАЮ

Руководитель ПНИ  
  
Д.И. Муромцев  
«30» декабря 2014 г.

ПРОГРАММНЫЙ КОМПЛЕКС ПРОГРАММНО-АППАРАТНОГО СТЕНДА

Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

ПАС.000000.001 00 01-ЛУ

Ответственный исполнитель  
  
М.А. Колчин  
«30» декабря 2014 г.

УТВЕРЖДЕН

ПАС.000000.001 00 01-ЛУ

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

ПРОГРАММНЫЙ КОМПЛЕКС ПРОГРАММНО-АППАРАТНОГО СТЕНДА

Текст программы

ПАС.000000.001 00 01

## АННОТАЦИЯ

В данном программном документе частично приведен текст программного комплекса ПАС, предназначенный для проведения экспериментальных исследований экспериментального образца масштабируемой сервис-ориентированной программно-аппаратной платформы для сбора, нормализации, обработки и визуализации больших массивов гетерогенных данных в распределенной сети электронных потребительских устройств.

Исходным языком программирования является Java. А средой выполнения является Java Virtual Machine 1.8 или выше.

Основные задачи программного комплекса:

- а) симулирует показания электронных потребительских устройств, таких как общедомовой счетчик тепловой энергии и уличный датчик температуры
- б) предоставляет доступ внешним системам к текущим моделируемым показаниям через программный интерфейс

## **К.1 Текст программного комплекса на исходном языке**

### **К.1.1 Текст подсистемы симуляции на исходном языке**

#### **К.1.1.1 Реализация механизма запуска**

```
package ru.semiot.simulator;

import java.util.ArrayList;
import java.util.List;
import madkit.kernel.Madkit;
import org.aeonbits.owner.ConfigFactory;

public class Launcher {

    public static void main(String[] args) {
        SimulatorConfig config = ConfigFactory.create(SimulatorConfig.class);

        List<String> params = new ArrayList<>();

        if(config.heatMeterAmount() > 0 || config.airTempAmount() > 0) {
            params.add("--launchAgents");
        }
        if(config.heatMeterAmount() > 0) {
            params.add("ru.semiot.simulator.HeatMeterAgent,false," +
config.heatMeterAmount());
        }
        if(config.airTempAmount() > 0) {
            params.add("ru.semiot.simulator.AirTempAgent,false," +
config.airTempAmount());
        }

        new Madkit(params.toArray(new String[]{}));
    }
}
```

#### **К.1.1.2 Реализация агента общедомового счетчика**

```
package ru.semiot.simulator;

import java.util.Random;
import madkit.kernel.Agent;
import madkit.message.StringMessage;
import org.aeonbits.owner.ConfigFactory;

public class HeatMeterAgent extends Agent {
```

```
    private static final String SHUTDOWN = "shutdown";
    private final Random random = new Random();
    private final SimulatorConfig config;
```

```

private final Database db;
private final HeatMeter meter;

private float heatAverage;
private float heatMax;
private float heatMin;
private float tempAverage;
private float tempMax;
private float tempMin;

public HeatMeterAgent() {
    this.config = ConfigFactory.create(SimulatorConfig.class);
    this.db = Database.getInstance();
    this.meter = db.newHeatMeter();
}

@Override
protected void activate() {
    if (logger != null) {
        logger.info("I'm ready to go!");
    }

    float[] heatAverages = db.getHeatAverage(config.month());

    heatAverage = heatAverages[0];
    heatMax = heatAverages[1];
    heatMin = heatAverages[2];

    float[] tempAverages = db.getHeatTempAverage(config.month());

    tempAverage = tempAverages[0];
    tempMax = tempAverages[1];
    tempMin = tempAverages[2];

    int pause = random.nextInt(config.heatMeterInterval());

    pause(pause);
}

@Override
protected void live() {
    while (true) {
        pause(config.heatMeterInterval());

        float heat = (heatMax - heatMin) * random.nextFloat();

        float temp = (tempMax - tempMin) * random.nextFloat();

        db.writeHeat(meter, heat);
        db.writeHeatTemp(meter, temp);
    }
}

```

```

        StringMessage msg = (StringMessage) nextMessage();
        if(msg != null && msg.getContent().equals(SHUTDOWN)) {
            break;
        }
    }
}

@Override
protected void end() {
    if (logger != null) {
        logger.info("Bye bye!");
    }
}
}

```

### **К.1.1.3 Реализация агента уличного датчика температуры**

```

package ru.semiot.simulator;

import java.util.Random;
import madkit.kernel.Agent;
import madkit.message.StringMessage;
import org.aeonbits.owner.ConfigFactory;

public class AirTempAgent extends Agent {

    private static final String SHUTDOWN = "shutdown";
    private final Random random = new Random();
    private final SimulatorConfig config;
    private final Database db;
    private final AirTemp airTemp;

    private float tempAverage;
    private float tempMax;
    private float tempMin;

    public AirTempAgent() {
        this.config = ConfigFactory.create(SimulatorConfig.class);
        this.db = Database.getInstance();
        this.airTemp = db.newAirTemp();
    }

    @Override
    protected void activate() {
        if (logger != null) {
            logger.info("I'm ready to go!");
        }
    }

    float[] tempAverages = db.getAirTempAverage(config.month());
}

```

```

        tempAverage = tempAverages[0];
        tempMax = tempAverages[1];
        tempMin = tempAverages[2];

        int pause = random.nextInt(config.airTempInterval());

        pause(pause);
    }

    @Override
    protected void live() {
        while (true) {
            pause(config.airTempInterval());

            float temp = (tempMax - tempMin) * random.nextFloat();

            db.writeAirTemp(airTemp, temp);

            StringMessage msg = (StringMessage) nextMessage();
            if(msg != null && msg.getContent().equals(SHUTDOWN)) {
                break;
            }
        }
    }

    @Override
    protected void end() {
        if (logger != null) {
            logger.info("Bye bye!");
        }
    }

}

```

#### **К.1.1.4 Интерфейс описывающий параметры конфигурации подсистемы**

```

package ru.semiot.simulator;

import org.aeonbits.owner.Config;

public interface SimulatorConfig extends Config {

    @Key("hibernate.connection.url")
    @DefaultValue("jdbc:mysql://localhost/phs-double ")
    String dbUrl();

    @Key("hibernate.connection.username")
    String dbUsername();

    @Key("hibernate.connection.password")
    String dbPassword();

```

```

    @Key("simulation.thermalmeter.amount")
    int heatMeterAmount();

    @Key("simulation.thermalmeter.interval")
    int heatMeterInterval();

    @Key("simulation.airtemp.amount")
    int airTempAmount();

    @Key("simulation.airtemp.interval")
    int airTempInterval();

    @Key("simulation.month")
    String month();

}

```

### **К.1.1.5 Файл сборки программы simulator.jar**

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.rspbench</groupId>
  <artifactId>madkit-1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-assembly-plugin</artifactId>
        <version>2.5.3</version>
        <executions>
          <execution>
            <id>make-assembly</id>
            <phase>package</phase>
            <goals>
              <goal>single</goal>
            </goals>
          </execution>
        </executions>
      <configuration>

```

```

<archive>
<manifest>
<mainClass>ru.semiot.simulator.Launcher</mainClass>
</manifest>
</archive>
<descriptorRefs>
<descriptorRef>jar-with-dependencies</descriptorRef>
</descriptorRefs>
</configuration>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.2</version>
<configuration>
    <showDeprecation>true</showDeprecation>
</configuration>
</plugin>
</plugins>
</build>
<dependencies>
<dependency>
<groupId>net.madkit</groupId>
<artifactId>madkit</artifactId>
<version>5.0.5.1</version>
</dependency>
<dependency>
<groupId>org.aeonbits.owner</groupId>
<artifactId>owner</artifactId>
<version>1.0.6</version>
</dependency>
</dependencies>

</project>

```

### **K.1.2 Текст подсистемы интеграции на исходном языке**

#### **K.1.2.1 Реализация СоAP –сервера на основе библиотеки Californium**

```

package ru.semiot.simulator.coap;

import java.net.SocketException;

import org.eclipse.californium.core.CoapResource;
import org.eclipse.californium.core.CoapServer;
import org.eclipse.californium.core.server.resources.CoapExchange;

public class SimulatorServer extends CoapServer {

```

```

private static final int PORT = 4545;

/*
 * Application entry point.
 */
public static void main(String[] args) {

    try {

        // create server
        SimulatorServer server = new SimulatorServer();
        server.addEndpoint(new CoAPEndpoint(PORT));
        server.start();

    } catch (SocketException e) {

        System.err.println("Failed to initialize server: " + e.getMessage());
    }
}

/*
 * Constructor for a new server. Here, the resources
 * of the server are initialized.
 */
public SimulatorServer() throws SocketException {

    // provide an instance of a resource
    add(new SensorInfoResource());
    add(new SensorObservationsResource());
}

```

### **К.1.2.2 Реализация обработчика запросов на получение информации об устройстве**

```

package ru.semiot.simulator.coap;

import org.eclipse.californium.core.CoapResource;
import org.eclipse.californium.core.coap.CoAP;
import org.eclipse.californium.core.server.resources.CoapExchange;

public class SensorInfoResource extends CoapResource {

    private static final String ID_PARAM = "id";
    private final RequestParser requestParser;
    private final Database db;
    private final RDFResponseBuilder responseBuilder;

```

```

public SensorInfoResource() {
    super("sensor/{id}");
    this.requestParser = new RequestParser(this);
    this.db = Database.getInstance();
    this.responseBuilder = new RDFResponseBuilder();
}

@Override
public void handleGET(CoapExchange exchange) {
    String id = requestParser.getPathParameter(ID_PARAM);

    if(id == null || !db.IsSensorExists(id)) {
        exchange.respond(CoAP.ResponseCode.NOT_FOUND);
    } else {
        Sensor sensor = db.getSensorById(id);
        exchange.respond(
            responseBuilder.createSensorInfoResponse(sensor));
    }
}
}

```

### **К.1.2.3 Реализация обработчика запросов на получения текущих показаний**

```

package ru.semiot.simulator.coap;

import org.eclipse.californium.core.CoapResource;
import org.eclipse.californium.core.coap.CoAP;
import org.eclipse.californium.core.server.resources.CoapExchange;

public class SensorObservationResource extends CoapResource {

    private static final String ID_PARAM = "id";
    private static final String NAME_PARAM = "name";
    private final RequestParser requestParser;
    private final Database db;
    private final RDFResponseBuilder responseBuilder;

    public SensorObservationResource() {
        super("sensor/{id}/{name}");
        this.requestParser = new RequestParser(this);
        this.db = Database.getInstance();
        this.responseBuilder = new RDFResponseBuilder();
    }

    @Override
    public void handleGET(CoapExchange exchange) {

```

```

        String id = requestParser.getPathParameter(ID_PARAM);
        String name = requestParser.getPathParameter(NAME_PARAM);

        if(id == null || name == null || !db.IsSensorExists(id)
        || !db.IsObservationExists(id, name)) {
            exchange.respond(CoAP.ResponseCode.NOT_FOUND);
        } else {
            Sensor sensor = db.getSensorById(id);
            Observation observation = db.getObservationBy(id, name);
            exchange.respond(responseBuilder.createSensorObservationResponse(
                sensor, observation));
        }
    }
}

```

#### **K.1.2.4 Файл сборки программы soap-server.jar**

```

<?xml version='1.0' encoding='UTF-8'?>
<project
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd"
    xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <modelVersion>4.0.0</modelVersion>

    <parent>
        <groupId>org.eclipse.californium</groupId>
        <artifactId>root</artifactId>
        <version>1.0.0-SNAPSHOT</version>
    </parent>
    <artifactId>cf-plugtest-server</artifactId>
    <packaging>jar</packaging>

    <name>Cf-PlugtestServer</name>
    <description>Californium (Cf) ETSI Plugtest server</description>

    <dependencies>
        <dependency>
            <groupId>org.eclipse.californium</groupId>
            <artifactId>californium-core</artifactId>
            <version>${project.version}</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-jar-plugin</artifactId>
                <version>2.2</version>

```

```

</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-assembly-plugin</artifactId>
<configuration>
  <appendAssemblyId>false</appendAssemblyId>
  <archive>
    <manifest>
      <addClasspath>true</addClasspath>
      <mainClass>org.eclipse.californium.plugtests.PlugtestServer</mainClass>
      <addDefaultImplementationEntries>true</addDefaultImplementationEntries>
    </manifest>
  </archive>
  <descriptorRefs>
    <descriptorRef>jar-with-dependencies</descriptorRef>
  </descriptorRefs>
  </configuration>
<executions>
  <execution>
    <id>make-assembly</id>
    <phase>package</phase>
    <goals>
      <goal>single</goal>
    </goals>
  </execution>
</executions>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-dependency-plugin</artifactId>
<executions>
  <execution>
    <id>copy-installed</id>
    <phase>install</phase>
  <goals>
    <goal>copy</goal>
  </goals>
  <configuration>
    <artifactItems>
      <artifactItem>

        <groupId>${project.groupId}</groupId>
        <artifactId>${project.artifactId}</artifactId>
        <version>${project.version}</version>
        <type>${project.packaging}</type>
      </artifactItem>
    </artifactItems>
    <outputDirectory>../run/</outputDirectory>
  </configuration>
</execution>

```

```

        </executions>
    </plugin>
</plugins>
</build>
</project>
```

### **K.1.2.5 Файл конфигурации системы логирования**

```

<configuration>
    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <!-- encoders are assigned the type
            ch.qos.logback.classic.encoder.PatternLayoutEncoder by default -->
        <encoder>
            <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36}
%msg%n</pattern>
        </encoder>
    </appender>
    <root level="info">
        <appender-ref ref="STDOUT" />
    </root>
</configuration>
```

### **K.1.2.6 Реализация генератора RDF данных на основе шаблона**

```

package ru.semiot.simulator.coap;

import java.io.IOException;
import org.apache.commons.io.IOUtils;

public class RDFResponseBuilder {

    private static final String OBSERVATION_TEMPLATE_NAME =
"observation.template";
    private static final String SENSOR_TEMPLATE_NAME = "sensor.observation";
    private static final String VAR_URI = "{URI}";
    private static final String VAR_VALUE = "{VALUE}";
    private String observationTemplate;
    private String sensorTemplate;

    public String createSensorInfoResponse(Sensor sensor) {
        try {
            if (sensorTemplate == null) {
                sensorTemplate =
IOUtils.toString(this.getClass().getResourceAsStream(SENSOR_TEMPLATE_NA
ME));
            }
            return new StringBuilder()
.append(sensorTemplate.replaceAll(VAR_URI, sensor.getUri()))
```

```
        .toString();
    } catch (IOException ex) {
        throw new IllegalStateException(ex);
    }
}

public String createSensorObservationResponse(Sensor sensor, Observation obsrv)
{
    try {
        if (observationTemplate == null) {
            observationTemplate = IOUtils.toString(this.getClass()
                .getResourceAsStream(OBSERVATION_TEMPLATE_NAME));
        }

        String response = observationTemplate.replaceAll(VAR_URI, sensor.getUri());
        response = response.replaceAll(VAR_VALUE, obsrv.getValue());
        return response;
    } catch (IOException ex) {
        throw new IllegalStateException(ex);
    }
}

}
```

**Лист регистрации изменений**

И з м .	Номера листов (страниц)				Всего листо в в докум енте	№ документа	Входящи й №	Подпись (фамилия)	Дата
	изме ненн ых	заме ненн ых	нов ых	анну лиро ванн ых					