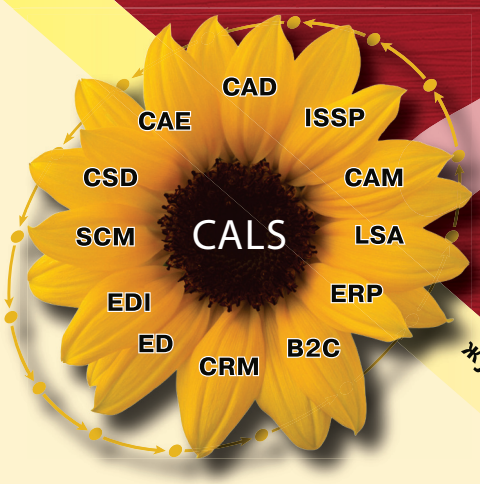
КАЧЕСТВО ИННОВАЦИИ ОБРАЗОВАНИЕ

№12
2014

ISSN 1999-513X



журнал в журнале

КАЧЕСТВО и ИПИ (CALS)-технологии

Качество • Инновации • Образование • №12-2014

www.quality-journal.ru

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

КАЧЕСТВО ИННОВАЦИИ ОБРАЗОВАНИЕ

№12 (115)
декабрь 2014

АДРЕС РЕДАКЦИИ И ИЗДАТЕЛЯ
105118, Москва, ул. Буракова, д. 8
Тел.: +7 (495) 916-89-29
Факс: +7 (495) 916-81-54
E-mail: quality@eqc.org.ru (для статей)
hg@eqc.org.ru (по общим вопросам)
www.quality-journal.ru; www.quality21.ru

●

ИЗДАТЕЛЬ
Европейский центр по качеству

●

НАУЧНЫЙ РЕДАКТОР
Гудков Ю.И.
ygudkov@hse.ru

●

ХУДОЖЕСТВЕННЫЙ РЕДАКТОР
Кудрявцева А.И.

●

ЛИТЕРАТУРНЫЙ РЕДАКТОР
Савин Е.С.

●

ОТВЕТСТВЕННЫЙ СЕКРЕТАРЬ
Мартюкова Е.С.
ne@eqc.org.ru

ЖУРНАЛ ЗАРЕГИСТРИРОВАН
в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций. Свидетельство о регистрации ПИ №77-9092

ПОДПИСНОЙ ИНДЕКС
в каталоге агентства «Роспечать» 80620, 80621
в каталоге агентства «Урал-Пресс» 14490
на сайте НЭБ eLIBRARY.RU 80620

ОТПЕЧАТАНО
ФГУП Издательство «Известия» УД ПРФ
127254, г. Москва, ул. Добролюбова, д. 6

Журнал входит в перечень ВАК РФ

Статьи рецензируются

СОДЕРЖАНИЕ

Сведения о членах редколлегии и об авторах статей можно найти на сайте www.quality-journal.ru

С.Г. Ефремов, Н.А. Пилипенко, Л.С. Восков

WEB ВЕЩЕЙ: ПРОБЛЕМЫ ПОИСКА И ВЗАИМОДЕЙСТВИЯ УСТРОЙСТВ

В настоящей работе рассматриваются проблемы поиска и организации взаимодействия устройств в рамках концепции Web'a Вещей. Производится обзор и анализ существующих решений в сфере обнаружения устройств, определяются ключевые требования, которым должны соответствовать поисковые сервисы. На основе стандартов EPCglobal предлагается расширение протокола поиска за счет дополнения его специализированным языком запросов. Для организации взаимодействия рассматриваются варианты адаптации существующих в настоящее время технологий асинхронной передачи, используемых для создания web-приложений, и на их основе предлагается создание адаптивного протокола, позволяющего учитывать динамически изменяющиеся требования устройств.

Ключевые слова: Интернет Вещей; Web Вещей; посредническая платформа данных; сервис обнаружения

1. Введение

Интернет Вещей (Internet of Things) [1] – это единая сеть, объединяющая в себе как виртуальные объекты, так и объекты реального мира. «Умная Вещь» (Интернет-вещь) – любой материальный или виртуальный объект, подключенный к сети Интернет. Концепция Интернета Вещей зародилась еще в 1999 году. С этого момента была создана целая экосистема так называемых интеллектуальных устройств, которые применяются практически во всех областях [2]: от небольших персональных носимых устройств, до полноценных систем Умных Городов. Согласно исследованиям Cisco [3], в 2008-2009 годах количество устройств, подключенных к интернету, превысило население Земли, что стало моментом перехода от «интернета людей» к «интернету вещей».

На начальном этапе развития стало формироваться множество проприетарных протоколов и технологий, используемых производителями в своем оборудовании, что привело к появлению совершенно несовместимых устройств. Ряд решений были предложены в качестве базовых протоколов: JINI, UPnP, DNLA, JXTA [4]. Появление WS – web-сервисов – привело к ряду исследований, направленных на их применение в рамках встраиваемых устройств и сенсорных сетей [5].

Однако постепенно все больше решений стало основываться на базовых web-технологиях, и появился термин Web Вещей [6]. Первые работы по использованию REST в Интернете Вещей [7] были нацелены на решение проблем повторного использования существующих web стандартов и технологий для организации доступа и взаимодействия между устройствами [7, 8, 9]. Дальнейшие работы были направлены на анализ COMET-технологий [10, 11] и стандартов HTML 5 [12].

Ключевые особенности Web'a Вещей:

- взаимодействие строится на базе протокола HTTP;
- каждое устройство представлено в сети сервисом с REST интерфейсом [8];
- интерфейсы устройств и систем открыты для интеграции с другими приложениями.

Свое дальнейшее развитие данное направление получило в виде концепции Социального Web'a Вещей (Social Web of Things – SWoT) [13, 14], в рамках которой устройства стали рассматриваться в качестве элементов «социального» взаимодействия, умеющих не только «общаться» друг с другом, но и иметь отношения, предоставлять свои ресурсы в доступ, и обладающих прочими возможностями, присущими современным социальным сетям. Подобный взгляд требует новых механизмов асинхронной связи между устройствами, так как существующие web-средства имеют другое предназначение.

Организации взаимодействия зачастую требуется обнаружение сервиса, являющегося виртуальным представлением данного устройства в сети. Поэтому множество работ были собраны вокруг сферы средств обнаружения [15] и, в частности, вокруг стандартов EPCglobal [16, 17]. Данные решения ориентированы в первую очередь на обнаружение сервисов конкретных объектов, т.е. по заранее определенным идентификаторам, и не предполагают альтернативных вариантов поиска.

Таким образом, целью настоящей работы является выполнение обзора и анализ существующих подходов к организации поиска и взаимодействия устройств в рамках Web'а Вещей, формирование списка ключевых требований и способов улучшения существующих решений.

2. Поиск и обнаружение устройств

Поиск и обнаружение устройств – одна из наиболее актуальных проблем Интернета Вещей. Увеличение количества интернет-ориентированных объектов совместно с развитием концепции открытых систем делает возможным создание глобальных динамических сценариев, включающих в себя сотни и тысячи одновременно функционирующих автономных устройств, взаимодействующих друг с другом для совместного решения задач.

Основная задача поиска состоит в выявлении необходимого источника информации или исполнительного устройства среди множества разнородных распределенных систем. При этом, критерии поиска могут включать в себя тип устройства, его географические координаты, функциональные возможности и прочие данные.

Как упоминалось ранее, на первом этапе большинство компаний использовали собственные проприетарные протоколы и технологии для организации взаимодействия между собственными устройствами и системами. Однако в настоящее время наблюдается отказ от таких решений и переход к созданию открытых систем на базе Web-технологий. При таком подходе задача поиска устройства сводится к поиску web-сервиса, являющегося представлением этого устройства в глобальной сети. Как показано в разделе 3, такие сервисы могут располагаться непосредственно на самих устройствах, а в случае, если они не имеют встроенных web-серверов – на специализированных шлюзах или посреднических платформах. Таким образом, каждое устройство получает свое виртуальное представление в сети Интернет, регламентируя лишь открытый интерфейс взаимодействия (RESTful API), но не определяя и не зная заранее, какие именно устройства или системы смогут взаимодействовать с ним в дальнейшем.

В сети предполагается наличие как минимум двух различных уровней сервиса (рис. 1).

Уровень сервисов устройств представлен всем множеством существующих устройств с открытыми web-интерфейсами.

Уровень агрегирующих сервисов включает различные мэшап-приложения, посреднические платформы, поисковые сервисы и пр. системы.

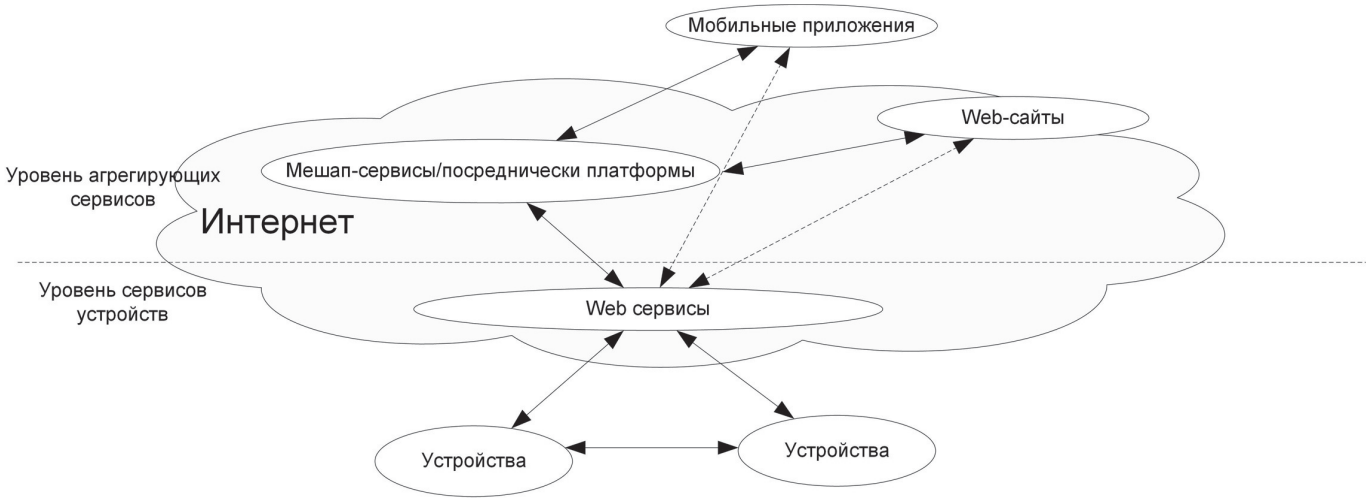


Рис. 1. Сервисы в Интернете Вещей

На сегодняшний день существует ряд решений по обнаружению объектов [15], наиболее значимыми из которых являются:

1. Частично утвержденные стандарты EPCglobal [18] представляют собой высокоуровневое описание архитектуры поискового сервиса. В основе предлагаемого подхода лежит EPC код

(уникальный для каждого устройства), EPCIS – сервис, предоставляющий данные для устройств с конкретным EPC, локальные поисковые сервисы, хранящие URI ссылки на все доступные EPCIS и центральный сервис EPCglobal.

2. BRIDGE представляет собой описание ряда подходов по реализации поискового сервиса. Ключевая идея – создание каталога сервисов. Каждые сервис регистрирует в нем информацию о доступности данных, соответствующих конкретному устройству. При этом, поисковый сервис выполняет функции оповещения заинтересованных клиентов об изменении данных. Запросы на получение данных также должны проходить через поисковый сервис к конкретному ресурсу.

3. Сервис Afilias, разработанный одноименной компанией, предназначен для решения сразу ряда проблем: идентификация объектов; обратная совместимость с уже существующими схемами идентификации; обеспечение практичности, масштабируемости и открытости решения; обеспечение высокого уровня безопасности и надежности. Для поддержки множества независимых схем используется ESDS (Extensible Supply-chain Discovery Service). Afilias представлен в виде прототипа с полной инфраструктурой, технической поддержкой и средствами разработки. Решение строится на базе DNS и совместимо с EPCglobal.

Помимо перечисленных выше, существует и ряд других проектов: ID@URI, DHT-P2P, WWAI, UDDI.

В связи с тем, что реализация динамического поиска устройств в глобальном масштабе практически невозможна, необходимо наличие специализированных сервисов обнаружения (Discovery Service – DS), агрегирующих в себе информацию об устройствах. Ниже представлены наиболее важные требования к таким сервисам:

1. Гибкая поддержка идентификаторов объектов (OID) (т.е. поддержка различных схем идентификации).
2. Запрос по OID должен возвращать весь набор серверов, имеющих данные об устройстве с соответствующим идентификатором.
3. Должна быть обеспечена возможность поиска не только по полному значению OID, но и по составным его частям, таким как номер партии, серия и пр., в случае если схема идентификации предполагает наличие таких составляющих.
4. Должна быть обеспечена возможность поиска по наличию конкретного атрибута.
5. Должна быть обеспечена возможность поиска по значению конкретного атрибута.
6. Может быть обеспечено хранение промежуточной информации для ускорения и упрощения схемы поиска.
7. Должен быть обеспечен полный набор функций для использования сервиса: регистрация устройства, аутентификация и авторизация.

Как было сказано ранее, стандарты EPCglobal включают базовое описание архитектуры Интернета Вещей, лежащее в основе большинства других решений. Однако в первую очередь данное решение направлено на обнаружение сервиса по заранее определенному коду (EPC) и не предполагает никаких альтернативных вариантов поиска устройства. Т.е. базовый сценарий работы определяет следующие шаги:

– обнаружение физического устройства;
– считывание его EPC, например, средствами RFID;
– обращение к EPCglobal для получения адреса сервиса (EPCIS);
– обращение к сервису для получения данных или управления.

Предлагаемое решение основано на данной архитектуре. Добавление языка запросов в такую систему может существенно расширить ее функциональные возможности. Это позволит обнаруживать объекты, которые могут находиться за пределами "видимости" конкретного устройства, а также необходимые сервисы, которые могут не иметь физического представления.

Для обеспечения такой возможности поисковый сервис должен представлять собой каталог, содержащий мета-описание и адрес каждого зарегистрированного ресурса (устройства).

Для поиска клиент отправляет GET запрос поисковому сервису с необходимыми параметрами, а сервис возвращает список устройств, удовлетворяющих запросу. Формат представления определяется в заголовке запроса. Сервис должен обеспечивать поддержку всех типовых форматов, таких как XML, JSON, CSV и пр.

Анализ ряда прикладных протоколов, основанных на HTTP, позволил выявить наиболее подходящее решение, которое может быть взято за основу – система фильтрации и поиска протокола OData V4 [19]. Ниже представлены главные функции, которые должны быть реализованы поисковым сервисом.

Запрос на поиск должен иметь следующий вид:

GET search?<filtering parameters>.

Например, запрос, возвращающий список устройств со значением атрибута Temperature равным 30, будет следующим:

GET search?$filter=Temperature eq 30

Ключевые параметры фильтрации должны включать в себя операторы сравнения, логические операторы, простейшие арифметические операторы, а также ряд дополнительных функций.

Операторы сравнения *eq, ne, gt, ge, lt, le* применимы для поиска по значению показателя, которое может быть равно, не равно, больше, больше или равно, меньше, меньше или равно значению поиска.

Операторы *and, or, not* используются для логической связки нескольких показателей.

GET search?$filter=Temperature gt 30 and Temperature lt 40

Арифметические операторы *add, sub, mul, div, mod* – для сложения, вычитания, умножения и деления.

GET search?$filter=(Temperature div 10) gt 3

Дополнительные функции, в свою очередь, могут включать в себя следующий набор:

– строковые функции *contains, endswith, startswith, length, indexof, substring, tolower, toupper, trim, concat.*

– функции для работы с датами и временем *year, month, day, hour, minute, second, date, time, now*.

– математические функции *round, floor, ceiling*.

– прочие функции: *hasproperty* (поиск по наличию свойства), *hasaction* (поиск по наличию команды), *distance* (вычисление расстояния между 2-мя координатами).

Примеры применения функций:

GET search?$filter= hasproperty('Temperature')

GET search?$filter= year (Updated) gt 2014

Параметры фильтрации могут также распространяться на комплексные типы данных:

GET search?$filter=Location.X eq 30.4

Протокол должен обеспечивать возможность постраничного показа, сортировки и выборки данных. Для этого должны использоваться такие операторы, как orderBy, top, skip:

GET search?$orderBy=Temperature desc

GET search?$top=10

GET search?$skip=20

Поиск в глобальном масштабе сервисом включает следующую последовательность действий:

1. Выбор всех устройств, соответствующих параметрам поиска.
2. Отправка запросов выбранным устройствам для проверки соответствия данных.
3. Возврат списка подходящих устройств клиенту.

Для минимизации взаимодействия между устройствами и сервисом, а также для уменьшения времени выполнения поиска поисковый сервис может обеспечивать промежуточное хранение данных, а сами устройства в таком случае должны указывать время жизни данных (TTL).

3. Взаимодействие устройств

Как было рассмотрено в [6], существует несколько сценариев по выводу устройств в Интернет.

3.1 Прямой доступ

При таком подходе прибор выполняет роль web-сервера. Вполне вероятно, что в ближайшем будущем большинство встраиваемых систем будут поддерживать стек протоколов TCP/IP (в частности, при помощи 6LoWPAN), и каждое устройство будет иметь свой IP-адрес и функционировать в качестве web-сервера, предоставляя RESTful API для дальнейшего взаимодействия. Однако это требует высоких производительных возможностей и не может быть реализовано на всех устройствах.

Существует множество решений на рынке, использующих данный подход. Например, Sun SPOT [9] включает в себя ряд датчиков (освещение, температура и т.д.) и имеет встроенный web-сервер, предоставляющий RESTful интерфейс.

3.2 Доступ через интернет-шлюз

Обеспечение прямого доступа является сложной задачей для встраиваемых систем, имеющих ограниченные ресурсы. Множество технологий, направленных на минимизацию энергозатрат конечных устройств, приводит к ограничению их производительности. Таким образом, реализация полноценного сетевого протокола на микроконтроллере становится практически невозможной. Обычно сенсорный узел реализует только 2 или 3 нижних уровня OSI, которые в большинстве случаев не совместимы с TCP/IP (например, IEEE 802.15.4, Bluetooth, ZigBee). Проблема решается использованием специализированных интернет-шлюзов, которые, с одной стороны, поддерживают протокол HTTP, а с другой – специализированные протоколы устройств. В таком случае только шлюз имеет собственный IP-адрес и функционирует как web-сервер.

Например, система интеллектуальных розеток Ploggs [9] состоит из беспроводных датчиков, измеряющих энергопотребление и работающих по протоколу Bluetooth, и центрального интернет-шлюза с установленным web-сервером, через который пользователи получают доступ к датчикам.

3.3 Посреднические платформы данных

Основной целью использования посреднических платформ данных является упрощение поиска, контроля, визуализации и обмена данными с различными устройствами. Данный подход предполагает наличие централизованного сервера или группы серверов, целью которых является хранение и предоставление данных, полученных от устройств. На основании таких платформ можно строить как информационно-измерительные системы, так и, например, осуществлять мониторинг состояния оборудования или контроль состояния здоровья пациентов.

При достаточной открытости платформы могут быть обеспечены возможности создания пользовательских сценариев взаимодействия интеллектуальных устройств.

На рынке существует ряд подобных платформ. Некоторые из них до сих пор представляют собой проприетарные решения, а некоторые, наоборот, строятся как открытые системы [20].

3.4 Асинхронный Web Вещей

Задача обеспечения обратной связи с устройством в Web'е Вещей для быстрого обновления или управления является довольно сложной и актуальной для всех перечисленных выше сценариев.

Исторически веб-стандарты были ориентированы на синхронное взаимодействие, согласному которому клиент отсылает запрос серверу и в ответ получает необходимые данные. Переход к Web 2.0 привел к появлению различных способов асинхронного взаимодействия на основе стандарта HTTP. В настоящее время большинство таких решений объединены общим названием COMET [11].

Наиболее очевидным из них является Polling. Клиент отсылает периодические запросы серверу, чтобы проверить, были ли новые данные, или нет. Очевидным недостатком является наличие большой избыточности такого подхода. Для его улучшения интервал между периодически повторяемыми запросами может меняться динамически, в зависимости от необходимой скорости реакции на события.

При Long Polling клиент отправляет запрос серверу, который удерживает данное соединение открытым до тех пор, пока не придут новые данные, и только после этого отправляет ответ клиенту. Данный подход похож на обычный Polling за одним исключением – интервалы между запросами – динамические, и зависят от частоты обновления данных на сервере. Недостатком является увеличение количества открытых соединений.

Forever Frame заключается в создании на странице скрытого iframe. Данный iframe пересылается как фрагментированный (chunked) блок (определяется HTTP заголовком), что позволяет динамически подгружать в него данные, пока открыто соединение. При возникновении некоторого события сервер передает новый элемент <script> с кодом, который должен быть выполнен на клиенте. При этом, выполнение содержимого данного блока на клиенте выполнится, как только он

будет загружен. Несмотря на то, что этот подход поддерживается всеми браузерами, он не может быть реализован на устройствах с ограниченными ресурсами.

При использовании способа Server-Sent Events (SSE), который является частью стандарта HTML5, клиент подписывается на события сервера, информацию о которых получает через постоянно открытое соединение, т.е. данный подход использует принципы, подобные подходу с фрагментированными данными.

Технология WebSockets, также являющаяся частью стандарта HTML5, представляется одним из наилучших решений. Это надстройка над протоколом HTTP, предоставляющая функциональность классических сокетов с дуплексной передачей данных. На текущий момент есть только черновики данного стандарта, его поддерживают не все прокси-серверы и браузеры.

Однако ни один из указанных способов не может быть полностью применим в Web'е Вещей по ряду причин. Во-первых, устройства часто имеют ограниченные ресурсы, таким образом, не могут быть реализованы протоколы, требующие разбора и выполнения кода JavaScript на микроконтроллере. Во-вторых, в традиционном web'е предполагается, что сервер не лимитирован в плане ресурсов и может поддерживать несколько средств/протоколов, в то время как в Web'е Вещей сервисы могут располагаться непосредственно на конечных устройствах, которые имеют свои предпочтения в плане выбора конкретных средств взаимодействия и лимитированы в ресурсах. В результате это приводит к требованию наличия двухстороннего соглашения по выбору протокола. Более того, интеллектуальные устройства могут иметь изменяемые требования к качеству сервиса в зависимости от ситуации и его текущего состояния.

Таким образом, требуется наличие адаптивного протокола взаимодействия, который будет обратно совместим с существующими в настоящее время средствами, но также будет обеспечивать возможность динамического выбора средств передачи данных. За основу такого протокола может быть взят подход, используемый в библиотеке SignalR [21], которая предоставляет средства динамического выбора транспорта в зависимости от возможностей клиента и сервера. SignalR имеет предустановленные приоритеты выбора: WebSockets, SSE, Forever Frame и Long Polling, что не предполагает возможности многокритериального выбора. Поэтому дальнейшая работа предполагает улучшение процедуры инициализации соединения в SignalR, которая должна обеспечить возможность выбора наиболее подходящего для обеих сторон протокола взаимодействия в зависимости от их функциональных требований и возможностей.

4. Заключение

В настоящей работе были рассмотрены две важные проблемы современного Web'а вещей. Во-первых, при возникновении динамических сценариев взаимодействия между устройствами, связи между которым не могут быть определены заранее, возникает необходимость в наличии специализированных механизмов поиска и обнаружения других устройств и их виртуальных представлений. Во-вторых, при непосредственном взаимодействии требуются средства асинхронной передачи данных, которые имеют разные показатели с точки зрения избыточности данных, скорости передачи и энергоэффективности. Использование адаптивного протокола может решить проблему динамически изменяющихся требований устройств.

Дальнейшие исследования предполагают работу в нескольких направлениях. Во-первых, в настоящей работе не были рассмотрены проблемы безопасности предлагаемых решений. Поисковые сервисы должны обеспечивать наличие механизмов для предоставления информации только авторизованным ресурсам. При реализации взаимодействия устройства также должны иметь возможность проверки прав своих «собеседников», для чего необходимы внешние сервисы аутентификации и авторизации, а также средства «подписи» отправляемых сообщений. Во-вторых, предлагаемое решение поиска требует дальнейшей детализации протокола и проверки в более глобальном масштабе. В-третьих, модель выбора транспорта требует определения взаимосвязи ключевых критериев и конкретных протоколов, создания многокритериальной модели выбора и реализации процедуры согласования между двумя сторонами взаимодействия.

Данное научное исследование (проект № 14-05-0064) выполняется при поддержке Программы «Научный фонд НИУ ВШЭ» в 2014/2015 гг.

Список литературы

1. Восков Л.С. Интернет вещей // "Новые информационные технологии". Тезисы докладов XX Международной студенческой конференции-школы-семинара. Москва, МИЭМ, 2012. С. 89-94.

2. Atzori L., Iera A., Morabito G. The Internet of Things: A survey // Computer Networks 54 (15), 2010. С. 2787-2805.

3. Evans D. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything // White Paper, Cisco Internet Business Solutions Group, April, 2011.

4. Traversat B., Abdelaziz M., Doolin D., Duigou M., Hugly J. and Pouyoul E. Project JXTA-C: Enabling a Web of Things // Proceedings of the 36-th Annual Hawaii International Conference on System Sciences, 2003. С. 282-290.

5. Priyantha N.B., Kansal A., Goraczko M. and Zhao F. Tiny web services: design and implementation of interoperable and evolvable sensor networks // Proceedings of the 6-th ACM Conference on Embedded Network Sensor Systems (SenSys). New York, NY, USA: ACM, 2008. С. 253-266.

6. Восков Л.С., Пилипенко Н.А. Web вещей – новый этап развития Интернета Вещей // Качество. Инновации. Образование. 2013. № 2. С. 44-49.

7. Guinard D. A Web of Things Application Architecture – Integrating the Real-World into the Web [диссертация]. ETH Zurich, 2011.

8. Guinard D., Trifa V., Wilde E. A resource oriented architecture for the Web of Things // Internet of Things (IoT), 2010 (Tokyo, Japan).

9. Guinard D., Trifa V., Pham T., Liechti O. Towards physical mashups in the Web of Things // Proceedings of the Sixth International Conference on Networked Sensing Systems, (INSS), 2009.

10. Duquennoy S., Grimaud G., Vandewalle J.-J. The web of things: interconnecting devices with high usability and performance // Proceedings of ICESS '09, HangZhou, Zhejiang, China, May 2009. С. 323-330.

11. Пилипенко Н.А. Технологии асинхронного взаимодействия в Web'е Вещей // Научно-техническая конференция студентов, аспирантов и молодых специалистов НИУ ВШЭ. Материалы конференции. – М.: МИЭМ НИУ ВШЭ, 2014. С. 143-144.

12. Si-Ho Cha, Yoemun Yun HTML5 Standards and Open API Mashups for the Web of Things // Computer Applications for Web, Human Computer Interaction, Signal and Image Processing, and Pattern Recognition Communications in Computer and Information Science, 342, 2012. С. 189-194.

13. Guinard D., Fischer M., Trifa V. Sharing Using Social Networks in a Composable Web of Things // Proceedings of the 1st IEEE International Workshop on the Web of Things (WoT 2010) at IEEE PerCom, Mannheim, Germany. С. 702-707.

14. Atzori L., Iera A., Morabito G., Nitti M. The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization // Computer Networks 56 (16), Nov. 2012. С. 3594-3608.

15. Evdokimov S., Fabian B., Kunz S. and Schoenemann N. Comparison of discovery service architectures for the internet of things // Proceedings of IEEE Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'10), Newport Beach, CA, 2010. С. 237-244.

16. Beier S., Grandison T., Kailing K., Rantzau R. Discovery Services – Enabling RFID Traceability in EPCglobal Networks // 13-th International Conference on Management of Data, 2006. С. 214-217.

17. Kurschner C., Condea C., Kasten O. and Thiesse F. Discovery Service Design in the EPCglobal Network – Towards Full Supply Chain Visibility // Proceedings Internet of Things (IOT 2008), Zurich, Switzerland, 2008, ser. LNCS 4952. Springer-Verlag, Berlin-Heidelberg, 2008. С. 19-34.

18. Стандарты EPCglobal. URL: http://www.gs1.org/gsmp/kc/epcglobal (дата обращения 30.09.2014).

19. Спецификация протокола OData 4.0. URL: http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part1-protocol.html (дата обращения 02.10.2014).

20. Weinberger M., Köhler M., Wörner D., Wortmann F. Platforms for the Internet of Things – An Analysis of Existing Solutions // 5-th Bosch Conference on Systems and Software Engineering (BoCSE). URL: http://cocoa.ethz.ch/downloads/2014/02/1682_20140212%20-%20Bocse.pdf (дата обращения 25.09.2014).

21. Библиотека SignalR. URL: http://signalr.net (дата обращения: 01.10.2014).

Ефремов Сергей Геннадьевич,
канд. техн. наук,
ст. преподаватель кафедры
Корпоративных информационных систем
факультета Бизнес-информатики НИУ ВШЭ.
e-mail: sefremov@hse.ru

Пилипенко Николай Александрович,
аспирант кафедры
Вычислительных систем и сетей,
НИУ ВШЭ.
e-mail: pilipenko-na@yandex.ru

Восков Леонид Сергеевич,
канд. техн. наук, профессор кафедры
Вычислительных систем и сетей,
НИУ ВШЭ.
e-mail: voskov@narod.ru

S.G. Efremov, N.A. Pilipenko, L.S. Voskov

WEB OF THINGS: PROBLEMS OF SEARCHING AND DEVICES COMMUNICATION

This article describes the problems of searching and devices communication in the Web of Things. It provides an overview and analysis of existing solutions in the field of discovery services and identifies the key requirements to them. It is proposed special query language for search protocol based on EPCglobal standards. After it is considered the adaption of the web technologies for asynchronous communication between devices and proposed the adaptive protocol to fit dynamically requirements of different devices.

Keywords: Internet of Things; Web of Things; middleware IoT platform; discovery service.

References

1. Voskov L. (2012) Internet veshhej [Internet of Things]. "Novye informacionnye tehnologii". Tezisy dokladov XX Mezhdunarodnoj studencheskoj konferencii-shkoly-seminara, M. MIEM, PP. 89-94.

2. Atzori L., Iera A., Morabito G. The Internet of Things: A survey // Computer Networks 54 (15), 2010. C. 2787-2805.

3. Evans D. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything // White Paper, Cisco Internet Business Solutions Group, April, 2011.

4. Traversat B., Abdelaziz M., Doolin D., Duigou M., Hugly J. and Pouyoul E. Project JXTA-C: Enabling a Web of Things // Proceedings of the 36-th Annual Hawaii International Conference on System Sciences, 2003. C. 282-290.

5. Priyantha N.B., Kansal A., Goraczko M. and Zhao F. Tiny web services: design and implementation of interoperable and evolvable sensor networks // Proceedings of the 6-th ACM Conference on Embedded Network Sensor Systems (SenSys). New York, NY, USA: ACM, 2008. C. 253-266.

6. Voskov L., Pilipenko N. (2013) Web veshhej – novyj jetap razvitija Interneta Veshhej [Web of Things – A new stage of the Internet of Things] // Kachestvo. Innovacii. Obrazovanie. № 2. PP. 44-49.

7. Guinard D. A Web of Things Application Architecture – Integrating the Real-World into the Web [диссертация]. ETH Zurich, 2011.

8. Guinard D., Trifa V., Wilde E. A resource oriented architecture for the Web of Things // Internet of Things (IoT), 2010 (Tokyo, Japan).

9. Guinard D., Trifa V., Pham T., Liechti O. Towards physical mashups in the Web of Things // Proceedings of the Sixth International Conference on Networked Sensing Systems, (INSS), 2009.

10. Duquennoy S., Grimaud G., Vandewalle J.-J. The web of things: interconnecting devices with high usability and performance // Proceedings of ICESS '09, HangZhou, Zhejiang, China, May 2009. C. 323-330.

11. Pilipenko N. (2014) Tehnologii asinhronnogo vzaimodejstvija v Web'e Veshhej [Asynchronous interaction technologies in the Web of Things]. Nauchno-tehnicheskaja konferencija studentov, aspirantov i molodyh specialistov HSE. Materialy konferencii. – M. MIEM HSE, PP. 143-144.

12. Si-Ho Cha, Yoemun Yun HTML5 Standards and Open API Mashups for the Web of Things // Computer Applications for Web, Human Computer Interaction, Signal and Image Processing, and Pattern Recognition Communications in Computer and Information Science, 342, 2012. C. 189-194.

13. Guinard D., Fischer M., Trifa V. Sharing Using Social Networks in a Composable Web of Things // Proceedings of the 1st IEEE International Workshop on the Web of Things (WoT 2010) at IEEE PerCom, Mannheim, Germany. C. 702-707.

14. Atzori L., Iera A., Morabito G., Nitti M. The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization // Computer Networks 56 (16), Nov. 2012. C. 3594-3608.

15. Evdokimov S., Fabian B., Kunz S. and Schoenemann N. Comparison of discovery service architectures for the internet of things // Proceedings of IEEE Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'10), Newport Beach, CA, 2010. C. 237-244.

16. Beier S., Grandison T., Kailing K., Rantzau R. Discovery Services – Enabling RFID Traceability in EPCglobal Networks // 13-th International Conference on Management of Data, 2006. C. 214-217.

17. Kurschner C., Condea C., Kasten O. and Thiesse F. Discovery Service Design in the EPCglobal Network – Towards Full Supply Chain Visibility // Proceedings Internet of Things (IOT 2008), Zurich, Switzerland, 2008, ser. LNCS 4952. Springer-Verlag, Berlin-Heidelberg, 2008. C. 19-34.

18. EPCglobal standards. URL: http://www.gs1.org/gsmp/kc/epcglobal.

19. OData 4.0 specification. URL: http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part1-protocol.html.

20. Weinberger M., Köhler M., Wörner D., Wortmann F. Platforms for the Internet of Things – An Analysis of Existing Solutions // 5-th Bosch Conference on Systems and Software Engineering (BoCSE). URL: http://cocoa.ethz.ch/downloads/2014/02/1682_20140212%20-%20Bocse.pdf (дата обращения 25.09.2014).

21. SignalR. URL: http://signalr.net

Sergey G. Efremov,
PhD, Senior Lecturer,
Department of Corporate Information Systems HSE.
e-mail: sefremov@hse.ru

Nikolay A. Pilipenko,
PhD student, Department
of Computer Systems and Networks HSE.
e-mail: pilipenko-na@yandex.ru

Leonid S. Voskov,
PhD, Professor, Department
of Computer Systems and Networks HSE.
e-mail: voskov@narod.ru