

Л.С. Восков, Н.А. Пилипенко

WEB ВЕЩЕЙ - НОВЫЙ ЭТАП РАЗВИТИЯ ИНТЕРНЕТА ВЕЩЕЙ

Интернет Вещей (Internet of Things - IoT) предполагает интеграцию предметов реального мира с компьютерными сетями. В данной статье рассматривается реализация этой концепции на прикладном уровне. Предлагается использование уже существующих архитектурных решений, ориентированных на разработку web-приложений. Производится обзор и анализ этих решений, выявляются их основные недостатки.

Ключевые слова: Интернет вещей, Web Вещей, REST, M2M

Глобальная сеть Интернет на сегодняшний день прочно вошла в нашу жизнь. Она используется для коммуникации, получения информации, общения, развлечения, обучения, ведения бизнеса. Большую популярность получили различные интернет-сервисы, предоставляющие возможность хранения и обработки информации "в облаке", тем самым, позволяя получать доступ к ней из любой точки планеты, с любого компьютера, ноутбука, коммуникатора и прочих устройств, имеющих доступ в Интернет.

Сегодня набирает обороты концепция Интернет Вещей. Данная концепция предполагает подключение к сети Интернет множества "умных вещей", взаимодействующих между собой. "Умная вещь", или Интернет-вещь, - это любой материальный объект, подключенный к сети Интернет.

Согласно [1], Web 3.0 - беспроводный Интернет вещей для коллективного создания и использования профессионального контента. Web - прикладной уровень сети, обеспечивающий взаимодействие пользовательских приложений с сетью. Он работает поверх Интернета, предоставляя пользователю интерфейс для получения данных в удобном для человека виде. Web 3.0 предполагает наполнение контентом и предоставление информации не только людям, но и всем устройствам, подключенным к сети.

Основной целью Интернета Вещей является предоставление доступа в сеть предметам реального мира. Соответственно, следующим шагом является переход на более высокий уровень - прикладной. Web Вещей (Web of Things - WoT) расширяет понятие Интернета Вещей, предполагая интеграцию умных вещей не только в сеть Интернет, но и в Web, т.е. на уровень приложений.

1. История развития

Создание в Интернете ссылок на объекты реального мира не является новой идеей. Изначально на предметах стали располагать специализированные маркеры (например, штрих-код или QR-код), которые позволяли пользователям через Интернет рассматривать информацию о них. Сегодня же различные маломощные устройства, месяцами работающие от батарейки, могут отправлять и получать данные через стандартные точки доступа

WiFi или интернет-шлюзы. Они могут напрямую взаимодействовать с "приборами на Web-странице", позволяя пользователю контролировать процесс измерения и управлять им с любого устройства, имеющего доступ к web-странице [2].

Самые ранние web-реализации данного подхода использовали HTTP-протокол только для передачи данных, несмотря на то, что HTTP - протокол, ориентированный на приложения. Соответственно, все усилия исследователей были сосредоточены на повторном использовании основных принципов, заложенных в Web. Основной целью являлась интеграция физических устройств с другим контентом в сети Интернет и создание единого языка для "общения" всех устройств.

Был предложен ряд проектов, обеспечивающих необходимую функциональность "умным вещам" для создания web-приложений. К ним относятся JINI, UPnP, DNLA и пр. Появление WS-* web-сервисов (SOAP и пр.) привело к созданию работ, связанных с их развертыванием на встраиваемых устройствах и сенсорных сетях. Однако они оказались слишком тяжеловесны для функционирования устройств. В результате было предложено использование REST (Representational State Transfer)-архитектуры [3], описывающей простой интерфейс управления информацией, не требующий специализированных API и дополнительных структур данных. К тому же данная архитектура поддерживается большинством современных социальных сетей, что упрощает их интеграцию.

2. Web-ориентированная архитектура

Web Вещей требует расширения возможностей существующей глобальной сети для легкой интеграции её с объектами реального мира и встраиваемыми системами. Вместо использования HTTP в качестве транспортного протокола (как реализуется в случае WS-* web-сервисов), предлагается интеграция устройств в качестве полноценной составляющей сети с использованием HTTP как протокола прикладного уровня.

По определению Роя Филдинга, REST - это набор общих принципов построения web-сервисов с определенными приоритетами: масштабируемость, независимость от платформы, расширяемость. Это описание архитектурного стиля, а не конкретный

ложений по обработке данных, предлагает открытый интерфейс и широкий набор инструментов для легкой разработки web-приложений.

EVERYTHING [15] позволяет интегрировать объекты с социальными сетями, CRM и прочими системами, предоставляет инструментарий для построения аналитических приложений.

MicroStrain's SensorCloud [16] - аналогичная платформа, изначально разрабатывалась для работы с MicroStrain датчиками. При помощи специализированного Windows-приложения SensorCloud Live Connect позволяет работать с устройствами через виртуальный последовательный порт, что существенно упрощает разработку приложений. Также предоставляет сервис MathEngine, который позволяет пользователям создавать скрипты для обработки данных.

Помимо описанных функций, все эти системы предоставляют API для взаимодействия на различных языках программирования (например, java, ruby), для различных платформ (arduino) и мобильных операционных систем (iOS, android).

Сейчас данные платформы применяются в системах домашней автоматизации, телемедицине, системах мониторинга окружающей среды, для контроля вычислительных ресурсов и пр. Например, платформа WoTKit используется для обработки данных, получаемых от пульсоксиметра [14]. Bluetooth датчики подключаются к специализированному интернет-шлюзу, который передает данные в WoTKit для облегчения мониторинга состояния пациента во время движения на транспорте.

Несмотря на все описанные достоинства и возможности, существует и ряд проблем, связанных с данным подходом.

Все существующие платформы имеют собственные форматы представления данных и описания объектов. К тому же открытым остаётся вопрос о порядке взаимодействия приборов с системой. Периодический опрос приборов для передачи данных требуется даже в случае отсутствия обновленных данных, что может снижать время функционирования устройств без внешних источников питания. Если же устройство должно являться инициатором начала передачи данных, тогда возникает проблема с возможностью управления прибором. Всё это существенно усложняет возможность интеграции различных систем между собой, поэтому сейчас необходимо решение проблемы стандартизации абстрактной модели объектов и порядка обслуживания интерфейсов различными платформами.

Поиск вещей в сети не сопоставим с поиском информации на интернет-страницах. Необходим динамический масштабируемый механизм поиска, учитывающий физику реальных устройств. Новейшие разработки в области семантического описания, которые могут быть встроены в HTML, такие как микроформаты или RDFa, могут способствовать решению данной проблемы.

В случае, когда производительность и время автономной работы имеют решающее значение

(например, когда устройства работают от батареи) использование оптимизированных протоколов, минимизирующих сетевое взаимодействие и задержки, будет оставаться наилучшим вариантом. Для решения данной проблемы может быть использована расширяемая архитектура, позволяющая разработчикам для взаимодействия с приборами использовать не только REST-архитектуру и HTTP-протокол, но и другие протоколы, ориентированные на конкретные устройства или группы устройств.

Ни одно из рассмотренных решений не учитывает также, что приборы могут передавать не только периодически изменяющиеся данные, но и потоки данных (например, видео). Соответственно, необходимо наличие протокола, обеспечивающего возможность потоковой передачи данных. Для этой цели могут быть использованы существующие средства доставки медиа контента, такие как метод прогрессивной загрузки (Progressive Download), работающий на основе HTTP, или же протокол RTSP (Real-Time Streaming Protocol). Однако с развитием HTML5 более перспективным выглядит использование технологии WebSocket, поддержка которой есть в большинстве современных web-браузеров. Данная технология ориентирована на web-приложения с интенсивным обменом данными, требовательных к скорости передачи данных, и, соответственно, является наилучшим решением поставленной задачи.

5. Требования к посредническим платформам данных

На основании рассмотренного материала и с учетом [14, 17] можно определить основной список требований к посредническим платформам данных. Простое подключение "вещей". Необходимо наличие открытого интерфейса для подключения любых приборов, удобного инструментария для разработки, а также набора готовых решений. Визуализация данных. В связи со сложностью данной задачи многие компании, например Google, предлагают широкий инструментарий для рисования графиков, диаграмм, карт.

Соответственно, для упрощения взаимодействия должно обеспечиваться соответствие форматов данных устройств и форматов данных, используемых инструментарием визуализации.

Обработка данных. Необходимо наличие инструментария для сбора, хранения и совместной обработки данных, а также возможность оповещения пользователей при наступлении необходимых событий.

Создание публичных объектов. Ключевой момент - позволить пользователям использовать "вещи", которые опубликовали другие пользователи. Соответственно, необходимо наличие гибкой системы разграничения прав доступа, возможности публикации и обмена данными между вещами.

Расширение функциональности. Поскольку инструментарий не может обеспечивать всю необходимую функциональность, важно поддерживать возможность подключения внешних компонент, по-

L.S.Voskov, N.A.Pilipenko

WEB OF THINGS - A NEW STAGE OF THE INTERNET OF THINGS

Internet of Things (Internet of Things - IoT) involves the integration of real-world objects with computer networks. This article describes the implementation of this concept at the application level. It is proposed to use existing architectural decisions, focused on the development of web-applications. Under review and analysis of these solutions, identifies their main weaknesses.

Keywords: Internet of Things, Web of Things, REST, M2M

References:

1. Voskov L.S. Internet of Things // "New information technologies." Abstracts of the XX International Student Conference and Summer School, M. MIEM, 2012 (89-94).
2. Kneller V.Yu. "Instrumental cloud" - the concept of functioning sensory systems based on Internet technology // Sensors and Systems. 2010. Number 8. Pp. 66-69.
3. Guinard D., Trifa V., Wilde E. A resource oriented architecture for the Web of Things // Internet of Things (IoT), 2010 (Tokyo, Japan).
4. Guinard D., Trifa V., Pham T., Liechti O. Towards physical mashups in the Web of Things // Proc. of the Sixth International Conference on Networked Sensing Systems, (INSS), 2009 (1-4).
5. Bufalino A., Spanghero S. // M2M in the Cloud: It's the Right Place, September, 2011 [of the electron source]. URL: (Date accessed 08.2012)
6. AirVantage M2M Cloud Platform [electronic resource]. URL: <http://www.sierrawireless.com/productsandservices/AirVantage.aspx> (date accessed 08.2012).
7. ThingWorx - a software platform that brings together the people, systems and devices [electronic resource]. URL: <http://www.thingworx.com> (date accessed 08.2012).
8. Axeda - cloud platform to connect things [electronic resource]. URL: <http://www.axeda.com> (date accessed 08.2012).
9. Cosm [electronic resource]. URL: <https://cosm.com> (date accessed 08.2012).
10. Open Sen.se [electronic resource]. URL: <http://open.sen.se> (date accessed 08.2012).
11. ThingSpeak [electronic resource]. URL: <https://thingspeak.com> (date accessed 08.2012).
12. Paraimpu [electronic resource]. URL: <http://paraimpu.crs4.it> (date accessed 08.2012).
13. Kansal A., Nath S., Liu J. and Zhao F. SenseWeb: an infrastructure for shared sensing // IEEE Multimedia, 14 (4), 2007 (8-13).
14. Blackstock M., Lea R. WoTKit: A Lightweight Toolkit for the Web of Things // Web of Things (WoT) workshop at Pervasive 2012, June 2012, New Castle, UK.
15. EVRYTHING [electronic resource]. URL: <http://evrythng.net> (date accessed 08.2012).
16. SensorCloud [electronic resource]. URL: <http://www.sensorcloud.com> (date inversion 08.2012).
17. Pilipenko N.A. Multi-platform remote access of equipment // "New information technologies." Abstracts of the XX International student Conference and Summer School, M. MIEM, 2012 (241-243).

Voskov Leonid Sergeevich,
Ph.D., Professor, Department of Computer Systems and Networks MIEM NRU HSE .
 Tel.: (495) 125-06-73, (910) 401-35-71
 e-mail: voskov@narod.ru

Pilipenko Nikolay Alexandrovich,
Students of the Department of Computer Systems and Networks MIEM NRU HSE .
 Tel.: (926) 143-32-67
 e-mail: pilipenko-na@yandex.ru