

ПРЕДСТАВЛЕНИЕ УСТРОЙСТВ ИНТЕРНЕТА ВЕЩЕЙ В ВИДЕ ВЕБ-СЕРВИСОВ И АВТОМАТИЗАЦИЯ ИХ ВЗАИМОДЕЙСТВИЯ¹

Федоров Роман Константинович

К.т.н., в.н.с., e-mail: dstu@icc.ru

Шумилов Александр Сергеевич

Аспирант, e-mail: dstu@icc.ru

Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской Академии наук, 664033, Иркутск, ул. Лермонтова, 134

Аннотация. В современном мире информационных технологий активно развивается аппаратное обеспечение и компонентная база интернета вещей, а также сервисо-ориентированный подход к выполнению вычислений. В данной работе предлагается подход представления устройств в виде веб-сервисов с последующим заданием их взаимодействия при помощи системы выполнения композиций сервисов в распределенной гетерогенной среде, разработанной в ИДСТУ СО РАН.

Данный подход апробирован при автоматизации управления офисным помещением и находится в процессе расширения своих возможностей.

Ключевые слова: интернет вещей, сервисо-ориентированный подход, композиции сервисов и устройств

Цитирование: Федоров Р.К., Шумилов А.С. Представление устройств интернета вещей в виде веб-сервисов и автоматизация их взаимодействия // Информационные и математические технологии в науке и управлении. 2019. № 1 (13). С. 105–113. DOI: 10.25729/2413-0133-2019-1-09

Введение. В современном мире информационных технологий активно развивается область интернета вещей – концепции, представляющей различные электронные устройства в качестве полноценных участников сети, взаимодействующих друг с другом. Доступ к участникам интернета вещей осуществляется посредством специальных программных интерфейсов, доступных по сети. Устройства могут быть как сложными электронными приборами, задействованными, например, в промышленном производстве и управлении городским автомобильным трафиком, так и элементарными электронными компонентами, например, термометрами и реле.

Наравне с увеличением количества различных устройств, подключенных к сети интернет, растет разнообразие различных интерфейсов, которые используются для получения доступа к устройствам. Используются как стандартные интерфейсы сервисов (WSDL, WPS), так и специально разработанные API, специфичные для каждого устройства в отдельности. Таким образом, существует проблема получения доступа и управления большим количеством устройств, имеющих разнообразные интерфейсы.

¹ Работа выполнена при финансовой поддержке РФФИ (грант № 16-07-00411_а, 16-07-00554_а, 17-57-44006-монг, 17-47-380007-р-а, 18-07-00758-а, Программы Президиума РАН №27 и Интеграционных программ СО РАН №77, ИИЦ СО РАН №4.1, а также ЦКП ИИВС ИРНОК)

Увеличивается количество устройств, подключенных к глобальной сети, но имеющих стандартные настройки безопасности, подверженные атакам извне, что определяет проблему контроля доступа к устройствам. В данный момент существует большое количество поисковых систем, специализирующихся на получении информации об умных устройствах с настройками безопасности по умолчанию.

Как результат появления множества устройств, решающих разнообразные задачи, существует проблема настройки их взаимодействия. Данная работа предлагает и реализовывает подход, когда умные устройства представляются в виде веб-сервисов, что позволяет абстрагироваться от конкретных интерфейсов устройств, настраивать к ним доступ и организовывать взаимодействие устройств в виде композиции сервисов, при этом используется разработанная в ИДСТУ СО РАН программная система выполнения композиций сервисов в гетерогенных распределенных средах для организации взаимодействия между устройствами.

1.1. Рассматриваемые устройства. Представленная работа фокусируется на устройствах интернета вещей. Одним из наиболее популярных типов устройств по их назначению являются устройства для организации умного дома. Различного рода датчики температуры, влажности, света, реле включения и выключения электропитания, герконы, элементы систем сигнализации вторжения становятся все более доступными и популярными среди пользователей. В целях упрощения построения взаимодействия между элементами производители устройств обычно выпускают программно-аппаратные средства, предоставляющие интуитивный интерфейс для объединения устройств и настройки их работы. Различные производители как создают свои собственные закрытые интерфейсы взаимодействия с устройствами [7], что заметно ограничивает конечных пользователей при выборе способа организации их взаимодействия, так и предоставляют открытый доступ к своим устройствам с использованием существующих открытых стандартов [13, 10].

Другим типом устройств по их назначению являются устройства для промышленных предприятий и организаций. По аналогии с системами умных домов существуют системы умных городов, решающих гораздо более сложные задачи и требующие гораздо более высокой квалификации пользователей, но решающих те же самые задачи, что и умные дома – оптимизация ресурсов (в том числе человеко-часов) при совершении повседневной деятельности. Примерами использования системы умных городов является оптимизация светофоров для управления плотностью автомобильного трафика [5], управление грузопассажирской инфраструктурой для обеспечения максимально быстрой и предсказуемой скорости доставки [9]. Принципы автоматизации и оптимизации деятельности посредством интернета вещей также применимы к промышленности (управления промышленными предприятиями), экологии (отслеживание состояния окружающей среды, предупреждение и смягчение последствий природных катаклизмов), сельскому хозяйству (автоматизированные системы полива, подачи удобрений в зависимости от параметров почвы) [14, 4].

Устройства, участвующие в интернете вещей, в основном используют сетевой протокол TCP/IP. В случае проводного соединения самым распространенным способом является стандартная витая пара. В случае беспроводного соединения распространено как использование Wi-Fi и Bluetooth, так и специализированных беспроводных стандартов с пониженным потреблением энергии ZigBee [15] и Z-Wave [11]. В данной работе

рассматриваются устройства, подсоединенные к сети интернет в проводном и беспроводном режиме.

1.2. Композиции устройств. Композицией называется набор взаимодействий различных устройств, причем набор задействованных устройств в рамках композиции может различаться при запусках в зависимости от промежуточных результатов вызова устройств внутри композиции. Условно способы задания композиций устройств можно разделить на две категории – графический и текстовый. Графический способ задания предполагает использование графических примитивов для определения взаимодействий и больше распространен в потребительских решениях. Текстовый способ предполагает непосредственное программирование взаимодействия между устройствами посредством использования специального API.

Графический способ задания используют наиболее популярные потребительские программно-аппаратные системы для управления устройствами, например, программные решения от Losant [12] и Flowhub [6]. Одним из наиболее популярных программных средств для задания композиций устройств в текстовом формате является платформа Node-RED [8].

В ИДСТУ СО РАН была разработана программная система, позволяющая задавать композиции сервисов в гетерогенных распределенных средах с помощью языка программирования JavaScript. Данный подход к заданию композиций позволяет обрабатывать промежуточные результаты выполнения сервисов стандартными средствами языка программирования или с помощью сторонних библиотек. Планирование выполнения сервисов, участвующих в композициях, а также обработка изменений в вычислительной среде (включение и отключение узлов, меняющийся уровень загруженности вычислительных узлов, неполадки с сетевым оборудованием и т.д.) производятся автоматически без участия пользователя.

Таким образом, данная работа предлагает и реализует подход к заданию взаимодействий между устройствами с использованием программной системы выполнения композиций сервисов в гетерогенной распределенной вычислительной среде. Предполагается, что устройства имеют веб-интерфейс, реализованный в виде веб-сервиса или REST API.

2.1. Программная система выполнения композиций распределенных сервисов.

На рис. 1 приведена схема программной системы выполнения композиций распределенных сервисов, реализованной в тесном взаимодействии с Геопорталом – многопользовательским веб-приложением, предоставляющим различные инструменты по обработке и визуализации пространственных данных [1]. Основные модули рассматриваемой программной системы для работы с композициями сервисов рассмотрены ниже. В рамках Геопортала композиции сервисов могут называться сценариями сервисов.

Система хранения данных (СХД) предоставляет каждому зарегистрированному пользователю определенный объем дискового пространства, который может быть использован для хранения входных или результирующих параметров запускаемых сервисов и композиций сервисов. Доступ к дисковому пространству предоставляется с помощью веб-интерфейса, который позволяет загружать и скачивать файлы, а также организовывать доступное дисковое пространство по аналогии с интерфейсами файловых систем персональных компьютеров.

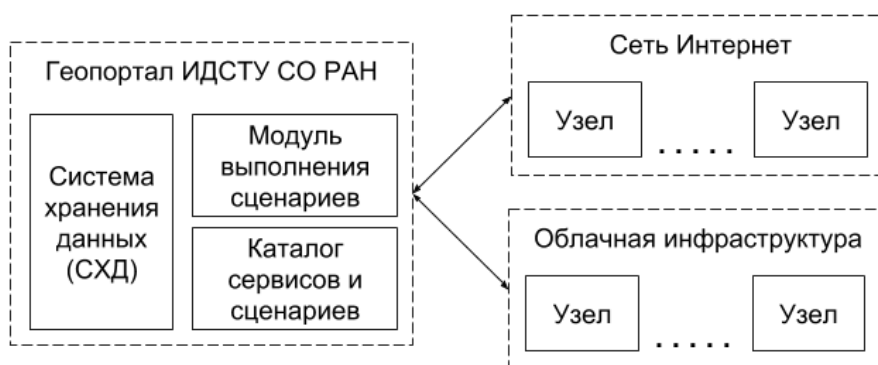


Рис. 1. Программная система выполнения композиций распределенных сервисов

Каталог сервисов и сценариев содержит информацию о зарегистрированных в системе сервисах и сценариях. Для того, чтобы вызвать сервис с помощью рассматриваемой системы, сначала необходимо произвести его регистрацию на Геопортале. Регистрация подразумевает под собой указание сетевого расположения сервиса, описание его входных и выходных параметров, а также его названия и описания. При описании параметров сервиса пользователь может определить элементы управления, которые будут использованы для ввода данных – например, если один из параметров сервиса является файлом, то элемент управления «выбор файла» перед запуском сервиса позволит загрузить или выбрать файл, находящийся на предоставленном пользователю дисковом пространстве. При регистрации сервиса пользователь должен указать как минимум один вычислительный узел, на котором развернут вычислительный сервис. Если указаны несколько вычислительных узлов, на которых развернут сервис, то решение о назначении вызова сервиса на определенный узел будет принято модулем выполнения сценариев.

Модуль выполнения сценариев осуществляет запуск и отслеживание выполнения сценариев и отдельных сервисов [2]. Данный модуль на вход принимает код сценария и, проанализировав его, автоматически составляет план выполнения сервисов, вызываемых в сценарии. Автоматическое распараллеливание выполнения сценариев вычислительных сервисов уменьшает время выполнения сценариев. Сценарии сервисов и устройств реализуются с помощью языка программирования JavaScript – сервисы и устройства представляются в виде стандартных функций, принимающих на вход параметры запуска сервисов и асинхронно возвращающие результаты выполнения. Особенности представления и вызова устройств будут рассмотрены ниже.

Вычислительные узлы, на которых развернуты вычислительные сервисы, могут быть расположены как в локальной облачной инфраструктуре, так и в любой точке сети Интернет – при регистрации сервиса на Геопортале достаточно указать публично доступный сетевой адрес вычислительного узла. Особенности локальной облачной инфраструктуры является наличие специальных программных инструментов, позволяющих пользователям Геопортала создавать виртуальные машины в рамках облачной инфраструктуры, которые уже имеют предустановленный набор программных средств разработки и публикации веб-сервисов [3].

Таким образом, Геопортал позволяет регистрировать сервисы и составлять из них композиции, используя язык программирования JavaScript. Подход к использованию языков программирования для задания композиций сервисов позволяет определять сложные

взаимодействия между сервисами (в случае переборных, рекурсивных, задач), а также обрабатывать промежуточные результаты выполнения сервисов прямо в сценариях стандартными средствами выбранного языка программирования. При выполнении композиций сервисов производится автоматическое планирование и распараллеливание выполнения вызова участвующих сервисов с учетом особенностей гетерогенной распределенной среде (изменчивый характер параметров среды, обработка возникающих критических ситуаций). Также реализуемый Геопорталом подход к заданию композиций сервисов обеспечивает возможность гибкой настройки доступа к сервисам.

2.2. Взаимодействие с устройствами. Данная работа рассматривает задачу применения программной системы выполнения композиций сервисов в распределенной гетерогенной среде для организации взаимодействия между устройствами. Для решения данной задачи необходимо дать определение устройствам в контексте решаемой задачи. Устройство – программно-аппаратная система, имеющая сетевой интерфейс и осуществляющая сбор информации об окружающей среде, либо оказывающая влияние на окружающую среду на основании команд, подаваемых извне, либо принимающая решения самостоятельно. Таким образом, задача организации взаимодействия устройств вырождается в задачу обеспечения возможности безопасного взаимодействия с устройствами внутри композиций (сценариев).

Работу с устройствами предлагается осуществлять посредством создания сценариев, внутри которых производится вызов устройств, используя стандартные средства JavaScript, нацеленные на взаимодействие с участниками сети, использующими HTTP протокол для соединения. Для этого в модуль выполнения сценариев интегрирован специальный JavaScript модуль, позволяющий совершать запросы к удаленным участникам сети по протоколу HTTP – используя GET, POST и т.д. вызовы. Данный модуль выбран по причине того, что большинство интерфейсов устройств представляют из себя веб-серверы (реализующих один из стандартов веб-сервисов или API).

В качестве примера будет рассмотрено создание сценария по управлению реле, отвечающим за включение и выключение света в помещении. Сценарий, приведенный ниже, сначала проверяет статус реле, затем, на основании переданных параметров, переводит реле в положение «включено» или «выключено» и возвращает результат выполнения. Входным параметром сценария является параметр `enable`, который может принимать истинное (включить) и ложное (выключить) значение.

```
function switch_light(input, mapping) {
  request.get("http://192.168.10.10/light", function(result) {
    if (result.status && input.enable || !result.status && !input.enable) {
      mapping.result.set(true);
    } else if (result.status && !input.enable || !result.status && input.enable) {
      request.post("http://192.168.10.10/light?key=123", function() {
        mapping.result.set(true);
      });
    } else {
      mapping.result.set(true);
    }
  });
}
```

2.3. Безопасность вызова устройств. Как уже было сказано ранее, одной из основных проблем при работе с устройствами является обеспечение безопасности их взаимодействия. Основным средством обеспечения безопасности и контроля доступа вызываемых сервисов является их абстракция в виде сценариев, используемая на Геопортале.

При создании сценария пользователь может определить круг пользователей, которые имеют право вызывать и редактировать сценарий. Если какой-либо пользователь системы имеет доступ только на вызов сценария, то он не может просмотреть его код, что делает невозможным получение информации о критических параметрах устройства (например, его сетевое расположение, способ вызова, использование секретных ключей, как, например, в примере выше).

Таким образом, устройство, регистрируемое в виде сценария, должно быть публичным (либо доступным только из сети, используемой Геопорталом, но это требует дополнительных настроек маршрутизации и настройки сетевого оборудования). Несмотря на публичность доступа к устройству, описание доступа к нему скрыто внутри кода сценария, что регулируется настройками безопасности. Данный вариант регистрации устройств на Геопортале оставляет возможность для случайного обнаружения устройства методом случайного перебора сетевых адресов и портов, но, при условии использования секретных ключей вызова вероятность получения несанкционированного доступа вследствие случайного обнаружения и перебора практически нулевая.

3. Апробация. В качестве апробации предлагаемого подхода к организации взаимодействия между устройствами с помощью программной системы выполнения композиций сервисов в распределенной среде приведен пример автоматизации управления световым оборудованием в офисном помещении предприятия.

В примере используется три устройства – реле включения и выключения света, сенсор уровня освещенности (возвращает показания освещенности по шкале от 0 до 1023) и датчик состояния сигнализации (возвращает булево значение статуса). Все устройства имеют веб-интерфейс и публичные сетевые адреса. Приведенный ниже программный код композиции запускается каждые 2 секунды и проверяет показания датчика освещенности посредством вызова `get_light_level()`. В случае если в настоящий момент сигнализация не включена (включение света в помещении, поставленном на охрану, может классифицироваться охранными датчиками как движение и вторжение) и если освещенность ниже порога `lightThreshold`, то производится включение света (переключение света происходит не чаще чем раз в 10 минут).

```
const lightThreshold = 400;
let lastChange = new Date();
let changeBuffer = 600000;
const smart_office = (input, mapping) => {
  setInterval(() => {
    get_light_level({}, {output: result1});
    get_security_status({}, {output: result2});
    if (result1.get() < lightThreshold && result2.get()) {
      let currentTime = new Date();
      let occurred = lastChange.getTime() - currentTime.getTime();
      if (occurred > changeBuffer) {
        lastChange = new Date();
        turn_on_light({}, {});
      }
    }
  });
}
```

```
}  
}  
}, 2000);  
}
```

Заключение. В статье показано, что существующая система выполнения композиций сервисов в распределенной гетерогенной среде была успешно использована для организации взаимодействия устройств интернета вещей. Сценарии взаимодействия устройств задаются с помощью языка программирования JavaScript, производится автоматическое распараллеливание вызова сервисов и адаптация к изменениям в вычислительной среде без участия пользователя.

Взаимодействие устройств осуществляется с учетом требований безопасности к использованию устройств (скрытие сетевого расположения устройств, параметров и способа вызова устройств, данных секретных ключей и т.д.) и максимально гибко в силу использования стандартных и любых сторонних средств и библиотек используемого языка программирования. Данный подход апробирован при автоматизации управления офисным помещением и находится в процессе расширения своих возможностей.

СПИСОК ЛИТЕРАТУРЫ

1. Федоров Р.К., Бычков И.В., Шумилов А.С., Ружников Г.М. Компоненты среды WPS-сервисов обработки геоданных // Вестник Новосибирского государственного университета. Серия: информационные технологии. 2014. Т. 12. № 3. С. 16–24.
2. Федоров Р.К., Бычков И.В., Шумилов А.С., Ружников Г.М. Система планирования и выполнения композиций веб-сервисов в гетерогенной динамической среде // Вычислительные технологии. 2016. Т. 21. № 6. С. 18–35.
3. Федоров Р.К., Шумилов А.С. Создание и публикация WPS-сервисов на основе облачной инфраструктуры // Вестник БГУ. 2015. № 4. С. 29–35.
4. Arooj M., Asif M., Shah S. Modeling Smart Agriculture using SensorML // (IJACSA) International Journal of Advanced Computer Science and Applications. 2017. Vol. 8. No. 5.
5. Celesti A., Galletta A., Carnevale L., Fazio M., Łay-Ekuakille A., Villari M. An IoT Cloud System for Traffic Monitoring and Vehicular Accidents Prevention Based on Mobile Sensor Data Processing // IEEE Sensors Journal. 2018. Vol. 18. No. 12. Pp. 4795–4802.
6. Flow-based programming for the full stack. Available at: <https://flowhub.io/> (accessed 07.07.2018).
7. Homekit – Apple Developer. Available at: <https://developer.apple.com/homekit/> (accessed 05.07.2018).
8. Node-RED | Flow-based programming for the Internet of Things. Available at: <https://nodered.org/> (accessed 06.07.2018).
9. Paprocki W. How Transport and Logistics Operators Can Implement the Solutions of "Industry 4.0" // TranSopot 2016: Sustainable Transport Development, Innovation and Technology. 2017. Pp. 185–196.
10. Sensor Model Language (SensorML) | OGC. Available at: <http://www.opengeospatial.org/standards/sensorml> (дата обращения: 6.07.2018).
11. The Internet of Things is powered by Z-Wave. Available at: <https://z-wavealliance.org/> (accessed 10.07.2018).

12. Visual workflow engine. Available at: <https://www.losant.com/iot-platform/visual-workflow-engine> (accessed 06.07.2018).
 13. Welcome | RAML. Available at: <https://raml.org/> (accessed 05.07.2018).
 14. Yang J., Liu M., Lu J., Miao Y. Botanical Internet of Things: Toward Smart Indoor Farming by Connecting People, Plant, Data and Clouds // *Mobile Networks and Applications*. 2017. Vol. 23. No. 2. Pp. 188–202.
 15. Zigbee Alliance. Available at: <http://www.zigbee.org/> (accessed 06.07.2018).
-

UDK 004.75

**IMPLEMENTATION OF SMART DEVICES AS WEB-SERVICES
AND AUTOMATION OF THEIRS INTERACTIONS**

Roman K. Fedorov

PhD., leading researcher, e-mail: dstu@icc.ru

Aleksandr S. Shumilov

Postgraduate, e-mail: dstu@icc.ru

Matrosov Institute for System Dynamics and Control Theory of
Siberian Branch of Russian Academy of Sciences, 664033, Irkutsk, Russia, Lermontov str., 134

Abstract. The active development of hardware and software of the internet of things devices, as well as the development of the service-oriented approach can be observed in today's world of informational technologies. This article proposes to implement various devices of the internet of things as web-services with definition of theirs interactions using the distributed service compositions execution in heterogeneous environment software system, developed in ISDCT SB RAS.

Presented approach was probated during automation of the office building management processes and is currently under constant development.

Keywords: internet of things, service-oriented approach, service and devices compositions

References

1. Fedorov R.K., Bychkov I.V., Shumilov A.S., Ruginikov G.M. Komponenty sredy WPS-servisov obrabotki geodannyh [Components of the WPS-service environment for spatial data processing] // *Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya: informacionnye tehnologii = Herald of Novosibirsk state university. Ser. Informational technologies*. 2014. Vol. 12. No. 3. Pp. 16–24. (in Russian)
2. Fedorov R.K., Bychkov I.V., Shumilov A.S., Ruginikov G.M. Sistema planirovaniya i vypolneniya kompozicij veb-servisov v geterogennoj dinamicheskoy srede [Software system of planning and executing web-services compositions in heterogeneous dynamic environment] // *Vychislitel'nye tehnologii = Computational technologies*. 2016. Vol. 21. No. 6. Pp. 18–35. (in Russian)

3. Fedorov R.K., Shumilov A.S. Sozdanie i publikacija WPS-servisov na osnove oblachnoj infrastruktury [Development and publication of WPS-services using the cloud infrastructure] // Vestnik BGU = Herald of BSU. 2015. No. 4. Pp. 29–35. (in Russian)
4. Arooj M., Asif M., Shah S. Modeling Smart Agriculture using SensorML // (IJACSA) International Journal of Advanced Computer Science and Applications. 2017. Vol. 8. No. 5.
5. Celesti A., Galletta A., Carnevale L., Fazio M., Łay-Ekuakille A., Villari M. An IoT Cloud System for Traffic Monitoring and Vehicular Accidents Prevention Based on Mobile Sensor Data Processing // IEEE Sensors Journal. 2018. Vol. 18. No. 12. Pp. 4795–4802.
6. Flow-based programming for the full stack. Available at: <https://flowhub.io/> (accessed 07.07.2018).
7. Homekit – Apple Developer. Available at: <https://developer.apple.com/homekit/> (accessed 05.07.2018).
8. Node-RED | Flow-based programming for the Internet of Things. Available at: <https://nodered.org/> (accessed 06.07.2018).
9. Paprocki W. How Transport and Logistics Operators Can Implement the Solutions of "Industry 4.0" // TranSopot 2016: Sustainable Transport Development, Innovation and Technology. 2017. Pp. 185–196.
10. Sensor Model Language (SensorML) | OGC. Available at: <http://www.opengeospatial.org/standards/sensorml> (дата обращения: 6.07.2018).
11. The Internet of Things is powered by Z-Wave. Available at: <https://z-wavealliance.org/> (accessed 10.07.2018).
12. Visual workflow engine. Available at: <https://www.losant.com/iot-platform/visual-workflow-engine> (accessed 06.07.2018).
13. Welcome | RAML. Available at: <https://raml.org/> (accessed 05.07.2018).
14. Yang J., Liu M., Lu J., Miao Y. Botanical Internet of Things: Toward Smart Indoor Farming by Connecting People, Plant, Data and Clouds // Mobile Networks and Applications. 2017. Vol. 23. No. 2. Pp. 188–202.
15. Zigbee Alliance. Available at: <http://www.zigbee.org/> (accessed 06.07.2018).