

УДК 004.031. 004.45

Средства и примеры разработки систем «интернета вещей»

Для разработки проектов «интернета вещей» (ИВ – IoT) используются различные стратегии, инструментари, платформы и технологии. Представлен анализ двух основных стратегий: гибридной и модельной. Рассмотрен инструментарий Mashup, который позволяет выполнять прототипирование IoT-разработки. Описаны три фазы разработки проекта на основе Mashup. Приведен состав и назначение компонент в пятиуровневой архитектуре разработки IoT. Даны элементы разработки IoT-прототипов с использованием плат: Arduino, Raspberry Pi, ESP8266 и Spark Core. Приведены критерии выбора подходящей платформы «интернета вещей» для разработки проекта IoT. Обсуждены примеры проектов ИВ и средства их разработки на российском рынке.

В. А. ВИШНЯКОВ,
д. т. н., профессор, профессор
кафедры ИКТ

Белорусский государственный
университет
информатики
и радиоэлектроники

Ключевые слова:
*сети IoT, стратегии,
инструментарий,
платформы, технологии,
примеры IoT-проектов,
качества обслуживания,
алгоритм обработки
очередей, класс трафика.*

Введение. Процесс разработки и внедрения систем «интернета вещей» (IoT) представляет собой последовательность взаимосвязанных мероприятий, начиная с этапа планирования проекта и заканчивая использованием системы. Структура программы по разработке IoT включает: определение цели; предварительные обследования и оценка ресурсов; стратегия решения IoT; выбор эталонной архитектуры IoT; дизайнерское решение; разработка мониторинга; внедрение системы; процедуры обеспечения качества; управление данными [1].

Процесс планирования можно разделить на три основных этапа. Первый этап включает в себя определение потребностей и определение целей IoT. После определения целей можно решить, какие данные необходимы и как они будут использоваться. Затем определяются стратегия решения, эталонная архитектура IoT и технологические платформы. Второй этап включает разработку проекта IoT. Третий этап включает в себя фактическую реализацию IoT-проекта.

Стратегии проектирования «интернета вещей». При разработке систем «интернета вещей» используются две основные стратегии: гибридная и модельная [2]. Гибридная предполагает, что система IoT разрабатывается путем объединения существующих сервисов и основана на известных инструментах и подходах веб-разработки (прототипировании), используется для некритичных приложений.

Подход, основанный на моделях, для проектирования и разработки систем на основе IoT был предложен в работе [2]. Также используется

инструментарий, например, язык моделирования «интернета вещей» (ThingML) [3], который предоставляет настраиваемую структуру генерации кода. Ее можно настроить для конкретных языков, промежуточного программного обеспечения, операционных систем, библиотек и систем [4].

В работе [5] определили два основных этапа методологии, основанной на модели: (1) идентификация требований и (2) формализация требований. Первый этап основан на консультациях и обсуждениях с экспертами предметной области для сбора информации о функциональности системы и идентификации требований. Функциональные требования могут быть представлены с помощью UML-диаграмм, описывающих функции системы IoT.

Второй этап направлен на структурирование требований и разработку моделей для описания взаимосвязи системных процессов. На этом этапе можно выделить подуровни с конкретными задачами, связанными с проектированием систем «интернета вещей», такими как координация, интеграция, управление большими данными и т. д. Основным преимуществом модельно-ориентированного подхода является платформенно-независимое моделирование, которое позволяет генерировать код для конкретных IoT-платформ.

Инструментарий проектирования. Инструменты мешапа (Mashup) позволяют выполнять очень быстрое прототипирование и использовать преобразование, преобразование и объединение данных из одной или нескольких служб для достижения

целей проекта. Они также позволяют подключать различные службы для создания новых процессов. Кроме того, некоторые инструменты мэшапа, такие как Clickscript [6], WotKit [7], Paraimpu [8], могут обеспечить средства моделирования и поддерживать взаимодействие между различными платформами. Инструменты могут быть эффективны при описании архитектуры системы, потока сообщений (диаграмм активности) и развертывания. Разработка на базе Mashup включает три фазы.

Фаза 1: Изучение среды развития для определения наиболее подходящих инструментов, доступных в настоящее время для достижения целей проекта IoT.

Фаза 2: Выбор платформы для интеграции между сервисами и удаленными платформами. Например, эта фаза может быть выполнена с помощью вычислительного ноутбука с открытым исходным кодом (Jupyter Notebook) [9].

Фаза 3: обработка данных и манипулирование ими. Сбор данных IoT может включать пять альтернативных протоколов сеансового уровня IoT: AMQP, CoAP, DDS, MQTT и XMPP.

Простейшая эталонная архитектура IoT включает в себя три уровня: датчик, сеть и приложение. Более сложная архитектура может содержать от пяти до семи уровней: устройство, сеть, обработка, приложение, бизнес, управление и безопасность.

Устройства IoT могут быть подключены либо непосредственно к сети, либо через шлюз, что позволяет устройствам взаимодействовать друг с другом, а также с облачными сервисами и приложениями [10]. Двигаясь вверх, компоненты становятся более сложными, а их связность увеличивается [11]. Рассмотрим состав и назначение компонент в пятиуровневой архитектуре IoT.

1. Датчики, исполнительные механизмы, интеллектуальные устройства и встроенные системы являются компонентами сенсорного слоя. Они собирают данные из различных физических, человеческих и природных сред и обеспечивают временное хранение этих данных. Они обеспечивают две основные возможности: сбор и анализ данных из окружающей среды.

2. Шлюзы IoT (концентраторы) – это устройства, которые соединяют компоненты первого слоя с сетями. Интегрированные в автомобильные двигатели, системы отопления, вентиляции и кондиционирования воздуха, используя полученные данные и их предварительную обработку, встроенные в концентраторы, позволяют конечным IoT-проектам адаптироваться к поведению среды и оптимизировать эффективность функционирования.

3. Сетевые и облачные сервисы обеспечивают инфраструктуру для функционирования IoT. Они могут

быть либо общедоступными (для населения), либо частично доступными (защищенными брандмауэром организации). Эти сетевые сервисы обеспечивают подключение к интернету концентраторов, а также предоставляют облачные вычислительные мощности, необходимые для сбора, хранения и анализа больших объемов данных с многих конечных устройств.

4. Центры обработки данных (облачные платформы) включает в себя наиболее технологически сложные компоненты «интернета вещей». Там выполняется интеллектуальная обработка полученных данных и принимаются соответствующие решения.

5. Расширенные сервисы (приложения) позволяют собирать и анализировать данные с различных платформ и предоставлять широкие интерактивные функции для конечных пользователей.

Сенсорные сети являются важным компонентом систем на основе «интернета вещей». Системы «интернета вещей» с обратной связью состоят из датчиков, центров обработки и исполнительных механизмов. Данные собираются с помощью датчиков. Затем эти данные через концентраторы, сенсорные сети передаются на облачные платформы, где обрабатываются, по результатам обработки принимаются соответствующие решения. Исполнительные механизмы выполняют системные операции на основе принятых решений.

Платы разработки и платформы «интернета вещей» для прототипирования. Во многих случаях процесс разработки IoT включает создание прототипов с использованием одной платы или их комбинаций, которые максимально соответствуют желаемой цели. Платы разработки IoT содержат стандартные средства связи, интерфейсы датчиков и соединения ввода-вывода общего назначения, так что разработчик может интегрировать датчики, исполнительные механизмы, средства связи и микроэлектромеханическую систему для создания прототипа своей сети IoT.

Самыми популярными платформами IoT для прототипирования являются Arduino, Raspberry Pi, ESP8266 и Spark Core [12]. Raspberry Pi идеально подходит для изучения серверных проектов Интернета вещей, может подключаться как к локальной сети, так и к сетям Wi-Fi. Однако не рекомендуется использовать Raspberry Pi для проектов, где есть изготовленная на заказ печатная плата, или для интеграции Raspberry Pi в готовый продукт.

Arduino идеально подходит для регистрации данных датчиков и управления исполнительными механизмами с помощью команд, отправленных на сервер другим клиентом.

ESP8266 лучше всего использовать в клиентских приложениях, таких как регистрация данных и управление исполнительными механизмами из онлайн-серверных приложений.

Платформа Spark Core идеально подходит как для серверных, так и для клиентских функций. Ее можно использовать для записи данных датчиков в облако Spark.io или для получения команд из облака.

Существуют основные критерии выбора подходящей платформы «интернета вещей» для проекта IoT [13]:

- тип подключения (Wi-Fi или сотовая связь) для системы IoT;
- тип услуги: некоторые услуги являются чисто платформами для подключения, а другие представляют собой комплексные решения, предлагающие оборудование, программное обеспечение и возможности подключения;
- срок службы проекта: как долго система «интернета вещей» будет работать;
- тарифный план: предлагает ли поставщик эффективный тарифный план;
- безопасность/конфиденциальность: как платформа IoT решает проблемы безопасности. Шлюзы облачной платформы должны предлагать шифрование SSL или DTLS;
- управляемая интеграция / доступ к API: как поставщик интегрирует сотовые модемы, сим-карты, диагностику устройств, обновления прошивки, облачные соединения, безопасность, уровень приложений и т. д.;
- избыточность и аварийное восстановление: есть ли у поставщика облачной платформы выделенная инфраструктура для обработки данных и выполняется ли резервное их копирование?
- экосистема IoT: отношения между сервисами платформы IoT;
- доступ к данным: соответствует ли услуга потребностям в интеграции данных, полученных через платформу IoT, с текущим облачным сервисом;
- аппаратное обеспечение: предлагает ли поставщик какие-либо готовые приложения, комплекты разработчика или стартовые пакеты для вашего варианта использования;
- управление устройствами: как поставщик позволяет отслеживать, сегментировать и управлять устройствами IoT;
- пограничный интеллект: платформа IoT должна иметь возможность расширяться от облака до тумана и поддерживать новые топологии для децентрализованных вычислений;
- обновления: как поставщик позволяет удаленно отправлять обновления и исправлять ошибки на устройствах.

Particle.io – это платформа, которая предлагает решения для разработки оборудования ИВ, возможности подключения, облако устройств и приложения. Particle.io производит широкую линейку

продуктов для разработки оборудования IoT как для прототипов, так и для производства на уровне DFM (design for manufacturing). Создание продукта ИВ в ней начинается с подключения устройств к интернету, и все платы микроконтроллеров Particle поддерживают связь по Wi-Fi, сотовой связи (2G/3G/LTE) или mesh-сети [14].

Средства разработки проектов ИВ на российском рынке [14], надо разделить два больших направления: ИВ-связь и ИВ-сервисы. ИВ-связь – это связь и сервисы, позволяющие этой связью управлять. Например, приобрели у оператора нужное количество SIM-карт, вставили их в M2M-устройства (счетчики, банкоматы или кассы) – и можно контролировать расходы и доступность устройств с помощью платформы M2M-мониторинга.

На этом сегменте рынка самыми емкими отраслями являются транспорт и логистика, хотя уровень проникновения довольно высокий. Также к ним можно отнести финансы и ЖКХ, развитие которого во многом стимулируется государством.

Рынок ИВ-сервисов реализуется иначе. Разрабатываются бизнес-приложения, которые работают с данными, собираемыми «вещами». Корректно собирать, обрабатывать, анализировать, визуализировать и, помогать автоматизировать процессы, настраивая логику работы «вещей» с участием или без участия человека. В каждой отрасли свои счетчики и датчики, свои «вещи», свои протоколы сбора и обмена данными, а главное – своя бизнес-логика работы.

Проект «Умное ЖКХ» собирает данные со счетчиков электричества, тепла, газа, воды, продукт предназначен для управляющих компаний и застройщиков. Решение «Онлайн-мониторинг воды» предназначено исключительно для водоканалов. И в отрасль логистики решения для ЖКХ не масштабируются – там другие датчики и другая бизнес-логика работы приложений.

Другой пример: у продукта «ВидеоАналитика» разработано более 20 сценариев работы компьютерного зрения: например, распознавать посетителей, считывать номер машины, чтобы открыть шлагбаум и т.д.

Самая емкая с точки зрения выручки и достаточно сложная для цифровизации с помощью ИВ-технологий – «Индустрия 4.0». Здесь требуется глубокая отраслевая экспертиза, знание производственных процессов, умение посчитать возврат инвестиций лучше, ведь ее проекты индивидуальные и дорогие.

С точки зрения развертывания технологии на инфраструктуре оператора связи, NB-IoT является привлекательной и перспективной технологией, так как использует инфраструктуру и оборудование уже существующей сети. Технология энергоэффективна

и работает в лицензируемом спектре, что гарантирует защиту от помех. Для ее развертывания требуются незначительные изменения в ядре сети и в ее радиочасти. Поскольку технология NB-IoT работает в LTE-стандарте, то для запуска потребовалось лишь обновить программное обеспечение и активировать технологию на сетевой инфраструктуре базовых станциях и опорной сети.

Заключение. Представлен анализ разработки проектов «интернета вещей». Для этого используются различные стратегии, инструментарии, платформы и технологии. Рассмотрены две основные стратегии разработки IoT-проектов: гибридная и модельная.

Проанализирован инструментарий Mashup, позволяющий выполнять прототипирование IoT-разработки, в котором используются три фазы проектирования на основе этих инструментов. Обсуждены состав и назначение компонент в пятиуровневой архитектуре разработки проектов IoT.

Процесс разработки IoT-прототипов выполняется с использованием плат Arduino, Raspberry Pi, ESP8266 и Spark Core. Приведены также критерии выбора подходящей облачной платформы «интернета вещей» для разработки проектов IoT. Рассмотрены примеры проектов ИВ и средства их разработки на российском рынке.

ЛИТЕРАТУРА

1. **Zhang, Y.**, Study on the Fire IOT Development Strategy / Y. Zhang, J. Yu // Procedia Engineering. - 2013. Vol. 52. - P. 314-319.
2. **Prehoferand, C.** From Internet of things mashups to model-based development / C. Prehoferand, L. Chiarabini // IEEE 39th Annual Computer Software and Applications Conference. - September 2015. - P. 499-504.
3. **Fleurey, F.** ThingML: A Generative Approach to Engineer Heterogeneous and Distributed Systems / F. Fleurey // IEEE International Conference on Software Architecture Workshops (ICSAW). - April 2017. - Pp. 185-188.
4. **Dias, J.** State of the Software Development Life-Cycle for the Internet-of-Things / J. Dias, H. Ferreira // 2019. [Electronic resource]. - Access mode: <https://arxiv.org/pdf/1811.04159>. - Access date: 25.12. 2021.
5. **Mezghani, E.** A Model-Driven Methodology for the Design of Autonomic and Cognitive IoT-Based Systems: Application to Healthcare / E. Mezghani, E. Expósito, K. Drira // IEEE Transactions on Emerging Topics in Computational Intelligence. - June 2017. Vol. 1 (3). -P. 224-234.
6. **Guinard, D.** A resource oriented architecture for the web of things / D. Guinard, V. Trifa, E. Wilde // Internet of Things (IOT). - November-December 2010. - P. 1-8.
7. **Blackstock, M.** IoT mashups with the WoTKit / M. Blackstock, R. Lea // 3rd International Conference on the Internet of Things (IOT). - October 2012. - P. 159-166.
8. **Pintus, A.** Paraimpu: a platform for a social web of things / A. Pintus, D. Carboni, A. Piras // 21st International conference companion on World Wide Web. ACM. - April 2012. - P. 401-404.
9. **Rule, A.** Exploration and Explanation in Computational Notebooks / A. Rule, A. Tabard, J. Hollan // CHI Conference on Human Factors in Computing Systems - CHI '18. - April 2018. - P. 1-12.
10. **Geber, A.** Simplify the development of your IoT solutions with IoT architectures / A. Geber, S. Cansal // IBM Developer. - August 2017. - P. 3-8.
11. **Burkit, F.** A Strategist's Guide to the Internet of Things / F. Burkit // 2014. [Electronic resource]. - Access mode: <https://www.strategy-business.com/article/00294?gko=a9303>. - Access date: 27.12. 2021.
12. **Lee, J.** How to Choose the Right IoT Platform: The Ultimate Checklist / J. Lee // Apr., 2018. [Electronic resource]. - Access mode: <https://hackernoon.com/how-to-choose-the-right-iot-platform-the-ultimate-checklist-47b5575d4e20>. - Access date: 30.12. 2021.
13. Top 10 selection criteria to choose your IoT platform. [Electronic resource]. - Access mode: <https://iotify.io/top-10-selection-criteria-for-your-iot-cloud-platform/>. Access date: 29.12. 2021.
14. Топ аппаратных платформ для проектов интернета вещей (IoT) 2021. [Электронный ресурс]. - Режим доступа: <https://mikrocontroller.ru/esp8266-projects/top-apparatnyh-platform-dlya-proektov-interneta-veshhej-IOT/> - Дата доступа: 15.09.2022.
15. «Интернет вещей» в России 2021. [Электронный ресурс]. - Режим доступа: https://www.cnews.ru/projects/2021/internet_of_things. - Дата доступа: 15.01.2022.

Various strategies, tools, platforms and technologies are used to develop Internet of Things projects (and in Ios). The analysis of two main strategies is presented: hybrid and model. The Mashup toolkit, which allow prototyping Ios development, is considered. Three phases of Mashup-based project development are described. The composition and purpose of components in the five-level architecture of IoT development is given. The process of developing IoT prototypes using boards: Arduino, Raspberry Pi, ESP8266 and Spark Core is given. The criteria for choosing a suitable Internet of Things platform for the development of an IoT project are given. Examples of S projects and means of their development on the Russian market were discussed.

Keywords: IT networks, strategies, tools, platforms, technologies, examples of IoT projects.

Получено 17.01.2022.