
УДК 004.415.2

В.А. ГОРБАЧЕВ, А.Ю. ЕФАНОВ

РАЗРАБОТКА МОБИЛЬНЫХ СЕРВИСНО-ОРИЕНТИРОВАННЫХ СИСТЕМ

Описывается разработка мобильной сервисно-ориентированной системы, предоставляющей доступ к ресурсам мобильного устройства через централизованную систему либо через другие мобильные устройства. Рассматриваются основные типы сервисно-ориентированных систем, а также общая архитектура системы, базирующаяся на облачных вычислениях.

Актуальность, новизна и цель работы

Постоянный рост вычислительных ресурсов мобильных устройств приводит к расширению возможностей их применения в повседневной жизни. В этой связи разработка приложений для них становится одним из основных направлений в индустрии программного обеспечения. Задача исследований в этой области – сделать рабочее место человека мобильным, предоставив ему соответствующие ресурсы для решения его проблем.

Современные мобильные устройства можно разделить на два типа:

– Непрограммируемые устройства. К данному типу можно отнести все мобильные либо встроенные (embedded) устройства, функции которых не могут быть расширены после производства, либо для этих устройств отсутствуют открытые программные интерфейсы.

– Программируемые устройства. К этому типу можно отнести устройства, обладающие открытыми программными интерфейсами, которые могут быть использованы для расширения функциональных возможностей устройства.

Для данного исследования интерес представляют устройства из второй группы, так как их функции могут быть расширены дополнительным программным обеспечением. Как правило, данные устройства имеют больше ресурсов центрального процессора, памяти, сетевой пропускной способности и, как следствие, могут стать сервисным провайдером. Провайдер сервисов означает, что устройство может предоставлять доступ к своим ресурсам через сеть различным клиентам, включающим другие мобильные устройства либо веб-браузер. К ресурсам мобильного устройства можно отнести как его аппаратные так и программные возможности, например, модуль позиционирования GPS, возможность записи звука, а также отправки текстовых сообщений и т.д.

На текущий момент использование ресурсов различных мобильных устройств в одном приложении распространено слабо. В данный момент начали распространяться сервисы для синхронизации данных между мобильными устройствами и компьютерами. Примерами таких сервисов являются Microsoft Live Mesh [1] и Apple Mobile Me [2]. Данные сервисы могут быть представлены как частный случай описанного ниже исследования.

Использование ресурсов различных мобильных устройств открывает большие перспективы для удобства работы, повышения производительности решения какой-либо задачи человеком, а также сокращения стоимости решения задачи.

К недостаткам такого подхода можно отнести зависимость от соединения с Интернетом. Соединение с сетью требуется для обмена данными между устройствами, а также системой управления ими. Возможность высокоскоростного соединения с Интернетом на текущий момент доступно не везде и требует дополнительной оплаты.

Цель работы – построение архитектуры мобильной сервисно-ориентированной системы, предоставляющей доступ к ресурсам мобильных устройств через централизованную систему.

Задачами исследования являются: 1) Анализ сервисно-ориентированных систем. 2) Разработка функциональной структуры мобильной сервисно-ориентированной системы. 3) Разработка программного обеспечения системы. 4) Оценка эффективности.

Рассмотрим задачи разработки функциональной структуры мобильной сервисно-ориентированной системы, а также ее программного обеспечения.

Общая архитектура системы

В данный пункт входит определение требований к системе, рассмотрение возможных вариантов использования системы в целом, определение спецификации требований к программному обеспечению на мобильном устройстве, а также требований к централизованной системе.

Одним из факторов, влияющих на архитектуру системы в целом, является определение эффективного метода взаимодействия между мобильными устройствами, а также между мобильным устройством и централизованной системой управления [3]. Существуют два метода, с помощью которых мобильные устройства осуществляют обмен данными:

Мобильное устройство как сервер

В этой модели мобильное устройство используется в качестве веб-сервера т.е. на устройстве физически размещается компонент, предоставляющий доступ к ресурсам устройства. В этом случае мобильные устройства устанавливают соединения друг с другом напрямую, а централизованная система используется для обнаружения устройств в сети, т.е. предоставляет адреса доступных устройств.

Преимущества: поскольку устройства устанавливают соединения напрямую, без функции управления, то производительность общего решения становится выше.

Недостатки: возможны проблемы с установкой сетевых соединений, так как большинство мобильных операторов имеют сетевую инфраструктуру, которая блокирует запросы из внешней сети к мобильному устройству.

Мобильное устройство как провайдер ресурсов

В этом случае клиентский компонент на мобильном устройстве устанавливает соединения с центральной системой и поддерживает его постоянно. Мобильное устройство обслуживает запросы, поступающие из системы, и посылает информацию о своем текущем статусе. Система транслирует запросы пользователя в запросы к мобильному устройству и имеет возможность управлять доступом к ним.

Преимущества:

- Система имеет возможность управлять доступом как к отдельным устройствам системы, так и к определенным ресурсам в системе.

- Управление устройствами и ресурсами мобильных устройств через веб-браузер более надежно, проще в реализации и обладает более высокой производительностью.

- Меньше проблем с установкой сетевых соединений с мобильными устройствами, так как устройство инициирует соединение и поддерживает его в активном состоянии.

Недостатки: более низкая производительность системы при работе с другими мобильными устройствами, так как весь сетевой обмен происходит через дополнительный сервер.

При дальнейшем проектировании системы была выбрана вторая модель из-за меньших проблем с организацией сетевого взаимодействия между устройствами и системой, а также чтобы иметь единое место, где контролируется доступ к устройствам.

Общая схема функционирования системы приведена на рис. 1. На схеме изображена центральная система управления, а также мобильные устройства, которые с ней взаимодействуют.

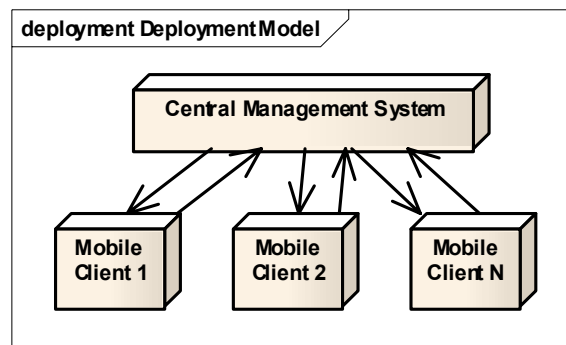


Рис. 1. Общая схема функционирования системы

Основными функциями центральной системы управления являются следующие:

- Хранение данных о мобильных устройствах, а также ресурсов, которыми устройство обладает.

- Обеспечение взаимодействия между устройствами и поддержка обмена сообщениями между ними.

- Поддержка приложений, использующих ресурсы доступных мобильных устройств.

- Обеспечение доступа к мобильному устройству и его ресурсам, а также к приложениям, использующим эти ресурсы через веб-интерфейс.

Среди основных функций мобильных клиентских приложений можно выделить следующие:

- Предоставление доступа центральной системы управления к ресурсам устройства.

- Предоставление пользовательского интерфейса, обеспечивающего доступ к ресурсам других устройств, а также к приложениям, использующим эти ресурсы.

Разработка программного обеспечения системы

В данную задачу входит определение конкретной реализации архитектуры системы, разработка детальных архитектур для каждого ее компонента, включающая клиентские приложения для различных мобильных платформ, а также ПО для централизованной системы, базирующейся на облачных вычислениях.

В качестве платформы для облачных вычислений была выбрана Windows Azure [4]. Данная система включает в себя ряд компонентов и дополнительных сервисов. Ядро системы базируется на платформе Windows и представляет собой кластер, объединяющий

большое количество серверов, на которых установлены виртуальные машины. Система доступна через единый интерфейс, так что весь кластер доступен пользователям как единая машина. Подобно другим платформам корпорации Microsoft данная поддерживает платформу .NET для различных типов приложений, включающих, как сайты на ASP.NET, так и WCF веб-сервисы.

Доступ к данным выделен как отдельный компонент и представляет собой веб-сервис, доступный через REST интерфейс [5]. Windows Azure позволяет хранить данные в нескольких форматах, таких как таблицы, очереди и бинарные объекты (blobs).

На рис. 2 приведена реализация архитектуры системы.

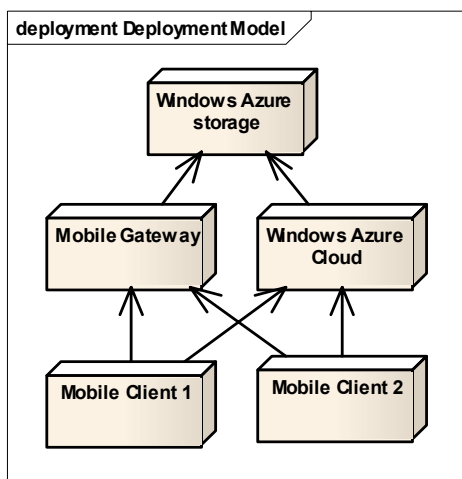


Рис. 2. Реализация архитектуры системы

В проекте используются следующие компоненты Windows Azure для хранения данных:

- Таблицы. В таблицах хранится вся доступная информация о системе, включающая данные мобильных устройств, их свойства, а также информация об интегрированных приложениях (т.е. приложениях, использующих ресурсы нескольких мобильных устройств).

- Очереди. Очереди используются для поддержки асинхронного взаимодействия между компонентами системы, такими как Мобильный шлюз (Mobile Gateway) и Обзорщик устройств (Devices Explorer).

Также на платформе облачных вычислений размещена централизованная система управления, которая использует следующие функции Windows Azure для своей работы:

- Компонент, выполняющий Веб-роль. Данный компонент представляет собой обычный веб-сервер, обрабатывающий HTTP запросы, на котором размещено веб-приложение, доступное через веб-браузер. Этот компонент реализует все функции, доступные пользователю, а также используется для размещения интегрированных приложений.

- Компонент с Рабочей ролью. Данный компонент используется для обработки данных, поступающих с мобильных устройств и их статуса. Компонент не имеет интерфейса пользователя и всегда работает в фоновом режиме. Взаимодействие с этим компонентом осуществляется через очереди либо через другие компоненты хранения данных Windows Azure.

Мобильный шлюз выполняет роль промежуточного сервера и используется для транслирования запросов мобильных устройств в запросы к подсистеме хранения данных. Мобильный шлюз предоставляет универсальный программный интерфейс (SOAP) для различных мобильных платформ. Этот компонент также используется для контроля доступа к данным и предотвращения несанкционированного доступа к ним. Компонент может быть размещен как на платформе облачных вычислений, так и в обычном окружении. В данном случае он размещен отдельно.

Мобильное клиентское приложение на рис. 1 представляет собой программное обеспечение для конкретной мобильной платформы. В рамках данного исследования были разработаны клиентские приложения для коммуникаторов Windows Mobile 6.1, а также iPhone OS 2.1, которые реализуют рассмотренные ранее основные функции мобильных клиентских

приложений. Клиентские приложения взаимодействуют с централизованной системой через мобильный шлюз, рассмотренный выше.

Реализация централизованной сервисно-ориентированной системы

Центральная система представляет собой приложение, размещенное на платформе облачных вычислений Windows Azure и состоящее из нескольких компонентов. Все компоненты системы реализованы с использованием возможностей платформы .NET Framework 3.5 на языке C#. Логически систему можно разделить на следующие компоненты.

- Компонент доступа к данным. Этот компонент инкапсулирует доступ к данным в системе Windows Azure и предоставляет программный интерфейс для доступа к таблицам, где хранятся данные об устройствах и их ресурсах, а также обеспечивает взаимодействие с очередями, которые используются для взаимодействия между мобильными устройствами и фоновым компонентом, рассмотренным ниже.

Данный компонент определяет несколько сообщений, которые могут быть помещены в очередь. Одно из них – это сообщение о состоянии устройства. Второе сообщение содержит информацию о текущем местоположении устройства, если она доступна.

Компонент также отвечает за проверку данных на соответствие допустимым значениям каждого поля.

Для обеспечения доступа к системе хранения данных о мобильных устройствах, их ресурсах, а также других данных системы были разработаны следующие классы.

- DeviceEntity – описывает объект устройства и его параметры.
- Devices – реализует логику добавления, удаления, изменения устройств, а также их выборку.
- DevicesDataContext – вспомогательный класс для представления коллекции объектов Devices в удобном для использования виде.
- DevicesQueue – класс для помещения и извлечения сообщений из очереди устройств.
- LocationAppQueue – класс для помещений и извлечения сообщений из очереди местоположений устройств.
- RuleViolation – класс для проверки корректности заполнения полей в классах Devices и Services.
- ServiceEntity – данный класс описывает структуру полей сервиса.
- Services – реализует логику добавления, изменения, удаления сервисов, а также их выборку.
- ServicesDataContext – вспомогательный класс для представления коллекции объектов.
- DevicePingMessage – класс для сериализации и десериализации сообщений в формат xml для последующего помещения их в очередь.
- Веб-приложение. Данный компонент представляет собой веб-приложение и предоставляет пользовательский интерфейс к системе и ее компонентам. Компонент работает в вычислительном облаке в качестве Веб-роли. Данное веб-приложение построено на основе библиотеки ASP.NET MVC 1.0. Библиотека основана на шаблоне проектирования MVC, где структура веб-приложения разделена на классы, относящиеся к модели данных, классы, определяющие представление, а также классы, обрабатывающие запросы пользователя и бизнес-логику.

Класс Account Controller отвечает за обработку запросов на авторизацию пользователей в системе и регистрацию пользователей в ней.

Класс Home Controller отвечает за обработку запросов к главной странице, содержащей ссылки на функции системы.

Класс Devices Controller предоставляет пользователю возможность управления своими устройствами, добавления новых устройств, их редактирования, а также определения текущего статуса устройств. Информация, представленная в пользовательском интерфейсе управления устройствами, приведена в таблице.

Класс Location App Controller реализовывает поддержку интегрированного приложения, показывающего текущее местоположение всех устройств, зарегистрированных в системе.

Класс MVC Application определяет привязку пользовательских HTTP запросов к конкретным классам контроллеров, рассмотренных выше.

Интерфейс списка устройств

Название	Статус	Модель	Время
WinMo	1	HTCTouch	6/28/2009 10:04
My iPod	0	Apple iPod	6/27/2009 09:44
Palm Win	0	Palm Win	6/25/2009 19:31

Приложение позиционирования

Для тестирования возможностей мобильной сервисно-ориентированной системы на реальной проблеме была рассмотрена задача определения глобального местоположения нескольких мо-

бильных устройств. Практическая ценность данного приложения в том, что оно может применяться для контроля перемещения грузов, спасательных операций, поиска и т.д.

Для решения данной задачи было разработано приложение, использующее модуль позиционирования GPS на нескольких мобильных устройствах. Приложение использует инфраструктуру центральной системы управления для получения доступа к данному ресурсу на всех мобильных устройствах и предоставляет пользовательский интерфейс для просмотра местоположения устройств на карте. Пользовательский интерфейс реализован как для мобильных клиентских приложений, так и для веб-браузера.

Приложение отображает текущее либо последнее местоположение всех зарегистрированных устройств на карте. Данные о текущем местоположении устройств клиентское приложение отправляет в систему каждые 5 секунд. Система показывает местоположение устройств как на платформе Windows Mobile, так и на платформе iPhone OS.

Приложение использует компонент доступа к данным для получения текущего местоположения устройств, а также Virtual Earth API для отображения данных на карте.

Это приложение также разработано на языке C# и имеет собственный контроллер обработки запросов пользователя.

Данные на карте показываются в режиме реального времени асинхронно, без необходимости обновления страницы. Пользователь также может увеличивать и уменьшать масштаб карты, что позволяет увидеть все устройства на земном шаре либо устройства на конкретной улице.

Фоновый компонент. Данный компонент всегда выполняется в фоновом режиме (рабочая роль) и не имеет пользовательского интерфейса. Компонент отвечает за обработку запросов с мобильных устройств и обновление информации об их текущем статусе и местоположении. Компонент работает с использованием очередей и получает данные для обработки из этого источника.

Мобильное устройство каждые 5 секунд посылает сообщение с информацией о текущем статусе и местоположении. Сообщение помещается в очередь для обработки фоновым компонентом.

Мобильное клиентское приложение. Это приложение представляет собой программу, реализованную для определенной мобильной программной платформы. Основная совокупность функций программы повторяет возможности веб-приложения Обозревателя устройств, однако данные доступны только в режиме для чтения.

Клиентское приложение предоставляет следующие функции:

- Список доступных устройств.

Приложение выводит на экран мобильного устройства список зарегистрированных устройств пользователя, а также их статус, указывающий на то, что устройство есть в сети и доступно для взаимодействия либо нет.

- Просмотр свойств определенных устройств. Клиентское приложение показывает различные свойства устройства, включая его модель и имя.

- Местоположение устройства. Приложение также показывает текущее местоположение выбранного устройства.

Все коммуникации с центральной системой управления происходят через промежуточный сервер, рассмотренный ранее, что позволяет предоставлять интерфейс, удобный для реализации на большинстве мобильных платформ и языках программирования.

Клиентское приложения для платформы Windows Mobile. Данное приложение было разработано с использованием .NET Compact Framework 3.5 и работает на платформе Windows Mobile 6.1. Приложение также было разработано на языке C#.

Клиентское приложение помимо коммуникации с мобильным шлюзом для получения данных об устройствах и регистрации в системе также использует веб-сервисы Virtual Earth для отображения местоположения устройств на карте.

Клиентское приложение для iPhone. Подобно мобильному клиентскому приложению для Windows Mobile было разработано также приложение для платформы iPhone OS 2.1. Данное приложение реализует весь набор функций, приведенный выше.

Основные функции данного приложения были реализованы на JavaScript. На языке Objective-C была разработана только логика запуска приложения.

Это приложение использует мобильный шлюз для взаимодействия с центральной системой управления, а также для получения данных. Однако для отображения устройств на карте применяются сервисы Google Maps ввиду их встроенной поддержки на данной платформе.

Внешний вид пользовательского интерфейса на карте отображен на рис. 3.

Выводы

В исследовании продемонстрирована возможность реализации мобильной сервисно-ориентированной системы на примере разработки функциональной структуры программного комплекса, построение его архитектуры в целом и его отдельных компонентов.

После определения структуры был реализован прототип централизованной системы управления на платформе облачных вычислений Windows Azure, а также разработаны мобильные клиентские приложения для платформ Windows Mobile и iPhone OS.

Также было разработано приложение просмотра местоположения мобильных устройств на карте, что демонстрирует концепцию интеграции ресурсов различных мобильных устройств в одном приложении.

Новизна исследования состоит в следующем:

- Разработан инструмент, который может использоваться для исследований в области сервисно-ориентированных систем, а также сервисов для мобильных устройств.
- Продemonстрирована возможность и эффективность интеграции мобильных устройств через централизованную систему.

Исследованы возможности применения платформы облачных вычислений для размещения масштабируемых, высокопроизводительных систем

Практическая ценность работы состоит в использовании мобильного устройства в качестве сервера, предоставляющего доступ удаленным клиентам к своим ресурсам через сеть.

Список литературы: 1. Microsoft Live Mesh, <http://mesh.com>. 2. Apple Mobile Me, <http://me.com>. 3. Gorton Ian, 'Essential Software Architecture' Springer, 2006. 356p. 4. Chappel D. 'Introducing the Azure Services Platform', 2008. 294p. 5. Richardson, Leonard Ruby, Sam, 'RESTful Web Services' (O'Reilly Media, Inc., 2007. 311p.).



Рис. 3. Местоположение устройства на карте

Поступила в редколлегию 20.10.2009

Горбачев Валерий Александрович, профессор кафедры ЭВМ ХНУРЭ. Научные интересы: моделирование систем. Хобби: музыка, волейбол, автомобили. Адрес: Украина, 61136, Харьков, ул. Героев Труда, 33, кв. 300, e-mail: Gorbachov Valeriy <gorbachov@kture.kharkov.ua> (Center NSIM).

Ефанов Александр Юрьевич, МСРД, ЭВМ ХНУРЭ. Научные интересы: сервис-ориентированные системы, компьютерные сети. Адрес: Украина, 61071, Харьков, ул. Кибальчича, 33, кв. 8.