

УДК 002.6:022

**МЕТОД ШАБЛОНОВ ПРИЛОЖЕНИЙ ДЛЯ ПОВЫШЕНИЯ
МОБИЛЬНОСТИ РАСПРЕДЕЛЕННЫХ СИСТЕМ СБОРА И
РЕТРАНСЛЯЦИИ ИНФОРМАЦИИ С БИОМЕДИЦИНСКИХ
ДАТЧИКОВ**

А. В. Петров, О. С. Большаков, А. С. Лебедев, Н. Е. Голубева

ООО «НаноАрм»

Получена 2 апреля 2013 г. после доработки 18 мая 2013 г.

Аннотация. В работе исследована проблема разработки кроссплатформенного программного обеспечения для распределенных систем сбора, ретрансляции и обработки данных от биомедицинских датчиков с помощью мобильных устройств на облачные сервера. Описан подход к решению проблемы на основе шаблона программного комплекса, приведено описание структуры предлагаемого решения и назначение каждого из компонент, указан порядок взаимодействия компонент шаблона.

Ключевые слова: мобильные приложения, инструменты разработки, кроссплатформенность, технологические платформы, компиляторы, трансляторы, облачные технологии, методологии разработки.

Abstract: We study the problem of developing cross-platform software for distributed systems for collecting, processing and relaying data from biomedical sensors on mobile devices for cloud servers. An approach to the problem of template-based software system, a description of the structure of the proposed solution and the function of each component, specify the method of component interaction pattern.

Key words: mobile application development tools, cross-platform, technology platforms, compilers, translators, cloud technologies, software development methodologies.

Введение

Сегодня мобильные вычислительные устройства (смартфоны) обладают значительной вычислительной мощностью, длительным временем автономной работы (до 4 вычислительных ядер общего назначения и 12 ядер специализированного назначения в процессоре Nvidia Tegra 3 пиковой производительностью до 12 Gflops [1]), широкой коммуникационной периферией, набором встроенных сенсоров. Можно сказать, что современные смартфоны и планшетные компьютеры прочно заняли нишу персональных вычислительных устройств [2] и число их пользователей постоянно растет, а статистические данные по интенсивности и продолжительности их использования одним пользователем сравнимы с обычными персональными компьютерами и ноутбуками.

Постоянно появляются новые области применения мобильных устройств. Одним из актуальных применений таких устройств является область дистанционного мониторинга состояния здоровья человека, за счет использования биомедицинских датчиков как источников данных, а мобильных устройств как вычислительных устройств предварительной обработки данных и ретрансляции. Примером может служить задача дистанционного холтеровского мониторирования ЭКГ человека.

В компьютерной индустрии и практике разработки программного обеспечения давно сложилась ситуация, при которой появление нового аппаратного обеспечения ставит перед разработчиками программ новые требования и вызовы. Аналогичную ситуацию мы можем наблюдать и в области разработки программного обеспечения для мобильных устройств. Требуются новые методы, модели, алгоритмы и инструменты, позволяющие эффективно создавать программное обеспечение для мобильных вычислительных устройств с учетом всех качеств, отличающих мобильные устройства от стандартных компьютеров. Требуется поиск новых подходов к проектированию и разработке кроссплатформенного программного обеспечения для мобильных систем.

Авторами данной работы был разработан экспериментальный образец программного комплекса «Транслятор данных», представляющий собой шаблон распределенной системы, позволяющей ретранслировать данные различного рода с помощью мобильных устройств и обрабатывать их на серверах по облачной модели предоставления услуг.

1. Шаблон распределенной системы ретрансляции данных с биомедицинских датчиков

Экспериментальный образец программного комплекса «Транслятор данных» (далее ЭО ПК) предназначен для экспериментальной реализации и проверки разработанных алгоритмов и методов, поддерживающих методику создания кроссплатформенных распределенных приложений ретрансляции и обработки данных с мобильными клиентами.

Прикладное назначение ЭО ПК заключается в сокращении сроков разработки кроссплатформенных распределенных приложений с мобильными клиентами, повышения качества разработанных приложений, за счет повторного использования функциональности ЭО ПК, реализованной на библиотечном уровне и за счет возможности автоматизированного добавления новой функциональности.

Таким образом, ЭО ПК реализует технологию автоматизированного проектирования и разработки высокопроизводительных конфигурируемых кроссплатформенных мобильных приложений с возможностями удаленного обмена данными с серверами приложений и серверными СУБД.

Отметим, что предлагаемый метод также предоставляет возможности для решения задач интероперабельности распределенных систем, однако в данной работе эта задача рассматриваться не будет.

Отметим, что задача разработки метода построения систем ретрансляции данных является актуальной, в том числе в области построения интегрированных облачных медицинских систем [3, 4, 5], поскольку её решение позволяет повысить актуальность получаемой от пациента

информации и сократить сроки реагирования на возникающие в его состоянии ситуации.

2. Структура предлагаемого решения

В состав разработанного ЭО ПК входят следующие модули:

- 1) модуль бизнес-логики;
- 2) модуль обмена данными с серверной СУБД и серверами приложений;
- 3) модули взаимодействия с периферийными устройствами, подключаемых или встроенных в МУ;
- 4) модуль аутентификации и авторизации пользователя мобильного приложения.

Данные модули реализуют необходимую базовую функциональность, требующуюся в системе сбора и ретрансляции данных в облачные сервера с помощью мобильных устройств (ретрансляторов). Ниже будет приведена обобщенная структура ЭО ПК (архитектура ЭО ПК) и указано положение каждого из вышеперечисленных модулей в общей структуре ЭО ПК.

ЭО ПК представляет собой расширяемый и конфигурируемый программный комплекс, включающий в свою структуру шаблон приложения, а также набор средств его конфигурации и адаптации под требования прикладной задачи.

Структура программного комплекса ЭО ПК представлена на рисунке 1.

В структуре ЭО ПК можно выделить следующие слои:

- 1) слой конфигурирования, включает в себя набор компонентов для конфигурирования, администрирования, расширения компонентов распределенной системы (мобильного приложения, серверного компонента и автоматизированного рабочего места с тонким клиентом);
- 2) слой шаблона приложения, включает в себя расширяемый библиотечный код повторно используемых компонентов, реализующих базовые системные функции по ретрансляции, хранению, защите данных, управлению рабочими процессами, распределению ресурсов, отображению информации;

3) слой пользовательских модулей, включает в себя определенный интерфейс расширения шаблона приложения, позволяющий подключать к приложению разработанные пользователями модули и тем самым расширять функциональность шаблона приложения. Данные модули реализуют клиентскую и серверную часть модуля бизнес логики, обозначенного выше.

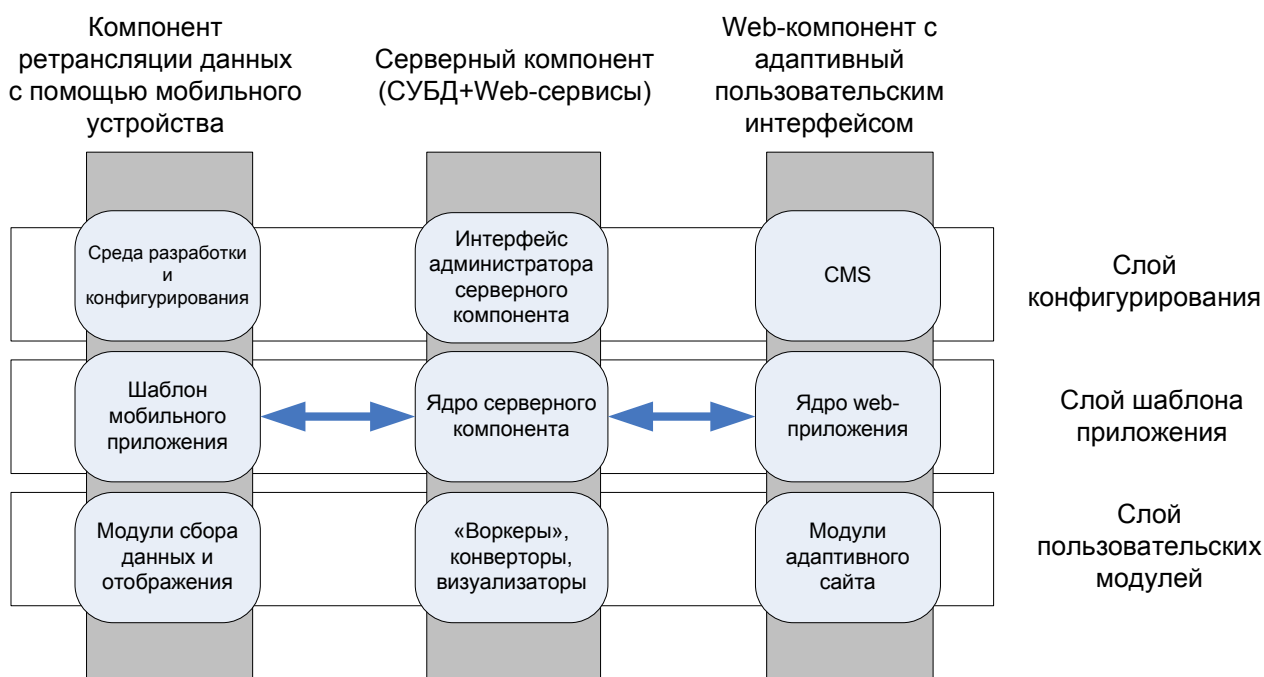


Рисунок 1 – Структура ЭО ПК

В структуре ЭО ПК можно выделить следующие компоненты:

- 1) компонент ретрансляции данных с помощью мобильного устройства, включающий в свою структуру шаблон мобильного приложения, средства его конфигурирования и средства расширения функциональности;
- 2) серверный компонент, включающий в свою структуру ядро, реализующее базовые распределенные сервисы, инструменты управления и администрирования серверного компонента, а также интерфейс для подключения пользовательских модулей;
- 3) web-компонент, включающий в свою структуру систему управления контентом, ядро приложения, реализующее API для связи с остальными

компонентами ЭО ПК, и интерфейс для подключения пользовательских модулей.

Назначение модулей компонента ретрансляции данных с помощью мобильного устройства заключается в следующем:

- 1) среда разработки и конфигурирования позволяет изменять шаблон мобильного приложения на уровне исходного кода на языке программирования C++ с использованием модулей автоматизированного проектирования и разработки (в частности с использованием модулей кодогенерации по UML-диаграммам и генерации реализации пользовательских протоколов с использованием разработанного метода генерации). Среда основана на платформе с открытым исходным кодом MoSync и обеспечивает кроссплатформенность создаваемых на базе шаблона мобильных приложений (поддерживаются ОС Android, iOS);
- 2) шаблон мобильного приложения кроссплатформенно реализует базовые сервисы ЭО ПК на языках C++ и Java на библиотечном уровне, позволяющие ретранслировать данные, хранить их на мобильном устройстве в случае отсутствия подключения к серверу, отображать диагностическую и другую информацию на экране мобильного ретранслятора; Шаблон мобильного приложения включает в себя реализованные модуль обмена данными с сервером (реализуются протоколы сетевого уровня модели OSI, протокол прикладного уровня генерируется по пользовательским описанием с использованием разработанного метода), модуль, предоставляющий возможность работы мобильного приложения в offline-режиме, модуль пользовательского интерфейса в «нативном» варианте реализации, модули регистрации и аутентификации; Данный компонент реализует модули обмена данными с серверной СУБД и серверами приложений и аутентификации и авторизации пользователя мобильного приложения, обозначенные выше.
- 3) модули сбора данных и отображения разрабатываются на языке C++ пользователем ЭО ПК в среде разработки и конфигурирования, назначение

модулей определяется решаемой прикладной задачей. Данные модули реализуют модули взаимодействия с периферийными устройствами, подключаемых или встроенных в МУ, обозначенные выше.

Назначение модулей серверного компонента заключается в следующем:

- 1) интерфейс администратора серверного компонента позволяет настраивать параметры ядра и пользовательских модулей;
- 2) ядро серверного компонента реализует базовые web-сервисы ретрансляции данных, балансировку обработки запросов к сервисам, планирование параллельного исполнения рабочих модулей («воркеров»), функции аутентификации и авторизации рабочих модулей и ретрансляторов, функции отказоустойчивости, функции дистанционного обновления мобильного приложения на мобильных ретрансляторах;
- 3) пользовательские модули серверного компонента, подключаемые к ядру серверного компонента через специфицированный интерфейс и конфигурируемые из административной части серверного компонента, реализуют на языке Java функции обработки ретранслируемых данных, при необходимости их визуализации и конвертации. Данные модули реализуют серверную часть модуля бизнес логики, обозначенного выше.

Назначение модулей web-компонента с адаптивным пользовательским интерфейсом заключается в следующем:

- 1) система управления контентом позволяет администрировать, конфигурировать и расширять web-приложение, в том числе за счет добавления нового статического и динамического контента;
- 2) ядро web-приложения, реализует на языке PHP шаблонный механизм генерации html-страниц по методу адаптивного пользовательского интерфейса web-страниц, за счет использования сформированных css и javascript файлов, а также SMARTY-конструкций описанных на языке XML. Ядро web-приложения включает в себя модули регистрации и аутентификации.

3) пользовательские модули, позволяют добавлять на языке программирования PHP новую функциональность в виде новых страниц web-приложения. Данные модули реализуют часть модуля бизнес логики, относящуюся к внешним по отношению к ЭО ПК приложениям и обозначенную выше.

3. Порядок взаимодействия составных компонентов

Взаимодействие компонентов ЭО ПК реализуется по схеме многоуровневого протокола взаимодействия. Все уровни, лежащие ниже прикладного уровня протокола взаимодействия (по модели OSI) реализованы в библиотечных API-функция шаблонов компонентов ЭО ПК [6, 7]. Реализация прикладного уровня протокола основывается на декларативном пользовательском описании, по которому согласно методу генерации реализации протоколов генерируется исходный код, интегрируемый с шаблонами трех основных компонентов ЭО ПК.

4. Анализ применимости разработанного метода

Приведем основные технологические подходы к разработке кроссплатформенных мобильных приложений [8 - 12]:

1) среды исполнения промежуточного представления (виртуальные машины, исполняющие промежуточный байт код и портированные под требуемые платформы);

2) генераторы нативного кода (трансляторы с промежуточного представления на язык, написанные на котором программы компилируются непосредственно под требуемую платформу, например: [2]);

3) веб-сервисы (web-приложения, использующие HTML5, Javascript, Web Socket и прочее);

4) нативные веб-обертки (приложения, скомпилированные под заданную платформу, но содержащие в себе браузер, позволяющий взаимодействовать из web-страницы с нативным приложением и тем самым упрощающий портирование графического интерфейса);

5) использование слоя виртуализации (например, проект [11], позволяющий запускать ОС Android на смартфонах под управлением Windows Mobile).

Данные подходы нацелены в первую очередь на обеспечение кроссплатформенности приложений для мобильных устройств, однако могут применяться и в случае персональных компьютеров и серверных станций.

Основным недостатком данных подходов является тот факт, что они не учитывают или учитывают не в полной мере, во-первых, распределенность создаваемых систем, а во-вторых, специфику систем ретрансляции данных.

Предлагаемый в данной работе подход, одновременно позволяет создавать компоненты, способные функционировать на различных платформах и в то же время решает вопросы, связанные с организацией эффективной архитектуры распределенной ретрансляции данных.

С точки зрения обеспечения кроссплатформенности предлагаемый подход отличается от существующих наличием специального слоя конфигурирования – инструментов позволяющих оптимально обеспечить поддержку той или иной требуемой платформы [13 - 16].

С точки зрения распределенности в предлагаемом подходе пользователю предоставляется набор готовых компонент, обеспечивающих базовые сервисы для распределенных систем. Предоставляемые компоненты являются шаблонами, поскольку дополняются пользовательскими модулями с использованием инструментов слоя конфигурирования.

Учет специфики систем ретрансляции обеспечивается использованием трехзвенной архитектуры с мобильными клиентами, качественно отражающей особенности систем ретрансляции [17 - 19], и использованием слоя пользовательских модулей, который позволяют описать бизнес-логику конкретной системы.

5. Выводы

В статье описан унифицированный подход к разработке распределенных систем сбора, ретрансляции и обработки данных с мобильными клиентами на

основе шаблона распределенной системы, позволяющий повысить такие качественные параметры системы как кроссплатформенность и интероперабельность её компонент.

Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации (госконтракт № 14.514.11.4030).

Литература

- [1] NVIDIA's Tegra 3 Launched: Architecture Revealed [Electronic Resource] / AnandTech. – Режим доступа:<http://www.anandtech.com/show/5072/nvidias-tegra-3-launched-architecture-revealed/2>, свободный. – Загл. с экрана.
- [2] Kramer D., Clark T., Oussen S. Platform Independent, Higher-Order, Statically Checked Mobile Applications // International journal of design, analysis and tools for circuits and systems. – 2011. – Vol. 2. – № 1.
- [3] Каменщиков А. А. Облачные технологии и интероперабельность информационных систем в здравоохранении / А. А. Каменщиков // Журнал радиоэлектроники. – 2013. – №2. – Режим доступа: <http://jre.cplire.ru/jre/feb13/11/text.html>.
- [4] Дабагов А. Р. Некоторые системные вопросы в контексте архитектур современных медицинских систем / А. Р. Дабагов // Журнал радиоэлектроники. – 2012. – №8. – Режим доступа: <http://jre.cplire.ru/jre/aug12/11/text.html>.
- [5] Дабагов А. Р. Информатизация здравоохранения и некоторые проблемы построения интегрированных медицинских информационных систем / А. Р. Дабагов // Журнал радиоэлектроники. – 2011. – №9. – Режим доступа: <http://jre.cplire.ru/jre/sep11/2/text.html>.
- [6] H. Zimmermann OSI Reference Model // The ISO Model of Architecture for Open Systems Interconnection, IEEE Transactions on communications, Vol. Com-28, No. 4, April 1980.
- [7] Бар Р. Язык Ада в проектировании систем: Пер. с англ. – М.: Мир, 1988.

- [8] J. Dehlinger, J. Dixon Mobile Application Software Engineering: Challenges and Research Directions // MobiCASE, Santa Monica, 2011
- [9] Т. С. Ткачева Исследование технологий кроссплатформенного программного обеспечения для мобильных устройств // Наука и техника ВВС Украины, 2011, #2(6), ISSN 2223-456X
- [10] D. Schlagwein Openness of Information Resources – A Framework-based Comparison of Mobile Platforms // 18th European Conference on Information Systems, 2010
- [11] D.R.S.Sai Krishna, G.Aparna, B.Sravani, Dr.K.Sreenivasa Ravi Cross Platform Compatibility on Mobile Operating Systems // IJCST Vol. 2, Issue 4, Oct. - Dec. 2011
- [12] M. Palmieri Comparison of cross-platform mobile development tools // Inderjeet Singh Malardalen University Innovation, Development and Technology, 2011
- [13] Particle Code / Particle Code. – Режим доступа: <http://www.particlecode.com/>, свободный. – Загл. с экрана.
- [14] А. П. Чувашов Технологии разработки мобильных приложений. плюсы и минусы разработки с помощью платформы Phonegap // IV Международная студенческая электронная научная конференция "Студенческий научный форум 2012" Пермский государственный национальный исследовательский университет, Пермь, Россия
- [15] Компания Lionbridge Информационное сообщение Mobile Web Apps vs. Mobile Native Apps: How to Make the Right Choice, 2012 <http://en-us.lionbridge.com/kc/mobile-web-apps-vs-mobile-native-apps.htm>
- [16] MoSync / MoSync. – Режим доступа: <http://www.mosync.com/>, свободный. – Загл. с экрана
- [17] Intel / Intel A Unified Mobile Architecture for the Modern Data Center – Режим доступа: <http://info.intel.com/rs/intel/images/IntelMobileArchitectureWhitePaper.pdf>, свободный. – Загл. с экрана.

- [18] O. Kassinen, E. Harjula, T. Koskela, M. Ylianttila Guidelines for the Implementation of Cross-platform Mobile Middleware // International Journal of Software Engineering and Its Applications, of Oulu, Oulu, Finland, Vol. 4, No. 3, July 2010
- [19] S. Tarkoma Mobile Middleware: Architecture // Patterns and Practice, Wiley, United Kingdom, 2009