

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА ОСНОВЕ ОНТОЛОГИЙ И СЕМАНТИЧЕСКИХ ДАННЫХ

Д. А. ЗАМУЛА, Д. И. МУРОМЦЕВ

*Университет ИТМО, 197101, Санкт-Петербург, Россия
E-mail: zamula.dmitry@gmail.com*

Предлагается новый подход к формированию функциональных и структурных элементов приложения, составляющих архитектурный уровень модели предметной области, за счет использования метаданных основного источника информации. Исходя из специфики требований к источнику данных предлагаемый подход наиболее актуален для систем, представляющих клиентскую часть распределенной информационной системы, где в качестве базы данных используются RDF-хранилища. На практике значительная часть подобных клиентов реализована в форме мобильных приложений.

Ключевые слова: онтология, предметно-ориентированное проектирование, мобильное приложение, RDF, SPARQL.

Введение. На сегодняшний день разработка мобильных приложений является одной из наиболее динамичных областей создания программного обеспечения. Существующая инфраструктура позволяет минимизировать затраты времени на реализацию общей функциональности клиентских приложений (работа с программными интерфейсами сервера, внедрение средств сбора аналитических данных и т.п.), что обеспечивает разработчикам возможность сосредоточиться на имплементации бизнес-логики, оперирующей объектами предметной области [1]. Данный этап является наименее формализованным, наиболее подверженным ошибкам и, как следствие, самым долгим в процессе разработки приложения.

В процессе создания мобильных клиентов можно выделить ряд особенностей, присущих данной области:

- быстрый переход от идеи к конечному продукту и, как следствие, короткие циклы разработки;
- потребность повторного использования частей кодовой базы при реализации общих компонентов в разных операционных системах;
- сфокусированность на одной предметной области;
- использование шаблонных решений для общих методов работы с данными (поиск, пагинация и т.п.).

Независимо от используемой методологии разработки в каждом проекте существует стадия построения модели предметной области [2]. Рассматривая наиболее широко используемую объектно-ориентированную парадигму программирования, можно определить данную стадию как процесс создания уровня бизнес-сущностей, представляющих собой структуры данных и описание их взаимоотношений (наследование и композиция). При необходимости данный уровень может содержать логику отображения структур основного хранилища информации (базы данных любых типов), выраженную в компонентах доступа к данным и промежуточных объектах, задействованных при трансформации структур хранилища в структуры бизнес-логики.

Для формирования архитектуры модели все чаще используется такой подход, как предметно-ориентированное проектирование. Данный набор принципов и паттернов построения приложений направлен на создание общего языка для описания сущностей, а также использование

техники разделения контекстов предметной области на изолированные компоненты (например, для социальной сети можно выделить компоненты аутентификации и авторизации, общения между пользователями, работы с коллекциями контента) [3].

Традиционный подход к разработке программного обеспечения. На рис. 1 представлены типичные фазы разработки прикладного программного продукта на примере приложения мобильного помощника в области культурного наследия, предоставляющего дополнительную информацию об объектах искусства (живописи), представленных на выставках. Рассматривая одну итерацию разработки, можно выделить следующие этапы, относящиеся к модели предметной области [4]:

- определение требований, формализованных в виде документации либо дополнительных знаний о предметной области у разработчика;
- описание сущностей в программном коде в виде классов и отношений между ними;
- реализация отображения программной модели на модель хранилища данных за счет выделенного уровня доступа;
- создание бизнес-логики приложения, включающей сервисный уровень манипулирования описанными сущностями, пользовательский интерфейс, компоненты интеграции со сторонними системами и т.п.



Рис. 1

В большинстве случаев, задача создания модели предметной области решается группой программистов вследствие использования единственного инструмента формализации модели — программного кода. При традиционном процессе разработки программист самостоятельно формирует модель, основываясь на имеющихся у него знаниях предметной области. Также возможно на этапе определения требований зафиксировать основные положения предметной области в документации, составленной аналитиками и экспертами предметной области, с целью дальнейшего перевода данных моделей в программный код. Из этого следует необходимость глубокого погружения разработчиков в предметную область, что реализуется посредством слабоформализуемого общения с экспертами.

Рассмотренный подход имеет ряд недостатков [5]:

- невозможность участия приглашенных экспертов в непосредственной формализации модели;
- необходимость дополнительных временных затрат на обмен знаниями между членами команды разработки;
- отсутствие инструментов поддержания архитектуры в непротиворечивом по отношению к предметной области состоянии: все архитектурные решения принимаются на основе текущего понимания разработчиком предметной области (потенциально неполного и противоречивого).

Описание знаний в виде программного кода ограничивает их применимость в рамках единого проекта, что обусловлено следующими факторами:

- модель приложения создается на основе текущих требований к конечной системе, а не реальных отношений в предметной области;
- существующие ограничения по трансформации исходного кода при использовании разных технологий разработки ПО для каждой мобильной платформы (различные языки программирования, библиотеки, фреймворки) не позволяют унифицировать описание модели;
- использование различных инструментов хранения данных может налагать дополнительные требования по формированию моделей (факторы производительности, атомарности запросов к хранилищу, средства поддержания непротиворечивости).

С учетом вышеуказанных особенностей разработки ПО, предназначенного для мобильных устройств, и недостатков традиционного метода конструирования приложений в настоящей статье предлагается новый подход к разработке мобильных клиентов с использованием предметно-ориентированной архитектуры, основанный на автоматизации построения уровня модели предметной области в бизнес-логике конечного приложения.

Генерация модели предметной области. Для решения обозначенных проблем предлагается выделить элементы модели предметной области в отдельный компонент, описываемый с помощью специализированных технологий работы со знаниями. Модель реального мира, используемая в приложении, непосредственно является набором формализованных в виде программного кода знаний, определяемых требованиями к ПО.

В качестве инструмента для работы со знаниями предлагается использовать стек технологий области семантического веба. В рамках описываемого подхода применяются следующие инструменты:

- онтологии, описывающие знания о предметной области (RDFS, OWL);
- формат представления данных (RDF);
- унифицированный интерфейс хранилища (язык запросов SPARQL).

Применяя язык OWL [6], возможно описывать знания о предметной области в виде отдельной структуры, не зависящей от программной платформы реализации мобильного приложения. В создании онтологии могут участвовать эксперты предметной области, используя специализированные программные средства для визуального редактирования онтологий [7]. При этом основное преимущество данного подхода выражается в назначении онтологии: при помощи данного средства описывается какая-либо область знаний без привязки к конкретному прикладному проекту, что позволяет использовать данное описание во множестве проектов, посвященных заданной предметной области. Таким же образом реализуется процесс повторного использования структур данных для мобильных приложений, предназначенных для разных программных платформ. При этом с учетом широких возможностей повторного применения онтологий конечному разработчику не требуется реализовывать их самостоятельно, так как на сегодняшний день множество предметных областей уже имеют онтологическое представление, и задача в данном случае состоит лишь в выборе подходящего.

Для использования онтологии в качестве базового средства проектирования архитектуры необходимы инструменты генерации модели предметной области в форме исходного кода требуемой программной платформы. Для решения данной задачи существуют специализированные инструменты перевода информации о структуре модели из онтологического представления в объектно-ориентированное [8]. При использовании языков программирования с динамической типизацией задача еще больше упрощается, сводясь к получению результата запроса к хранилищу информации в семантическом виде (триплстору).

Рассмотрим пример JavaScript-кода из приложения MobileCultureCloud — получение списка всех произведений одного автора:

```

var ALL_AUTHORS_WORKS = "SELECT DISTINCT * WHERE {
VALUES ?author { <%authorUri> }
?prod ecrm:P14_carried_out_by ?author .
?prod ecrm:P108_has_produced ?uri .
OPTIONAL { ?prod ecrm:P4_has_time-span ?date .
?date rdfs:label ?creationDate FILTER(
langMatches(lang(?creationDate), 'ru')) }
?uri ecrm:P138i_has_representation ?img .
?uri ecrm:P102_has_title ?titleInd .
?titleInd rdfs:label ?title FILTER(langMatches(lang(?title), 'ru'))
}";
getPaintings: function (authorUri, callback) {
$http.get(SPARQL_ENDPOINT +
ALL_AUTHORS_WORKS.replace(/%authorUri%/g, authorUri)).success(function
(response) {
callback(response);
});
};
    
```

В данном случае вся бизнес-логика (фильтрация) представлена в SPARQL-запросе, при этом функциональность представления полученных данных реализуется в функции “callback”.

Практическая реализация. На основе предложенного подхода было реализовано мобильное приложение MobileCultureCloud — клиент для информационной системы музея, предоставляющий пользователям дополнительную информацию о выставленных экспонатах, а также их авторах.

Архитектурная модель созданного приложения базируется на используемой онтологии хранилища информации CIDOC CRM [9]. На рис. 2 схематично представлена часть элементов данной модели с их отношениями в используемой онтологии. На рис. 3 показан скриншот приложения, явным образом визуализирующий информацию, содержащуюся в полученной модели.

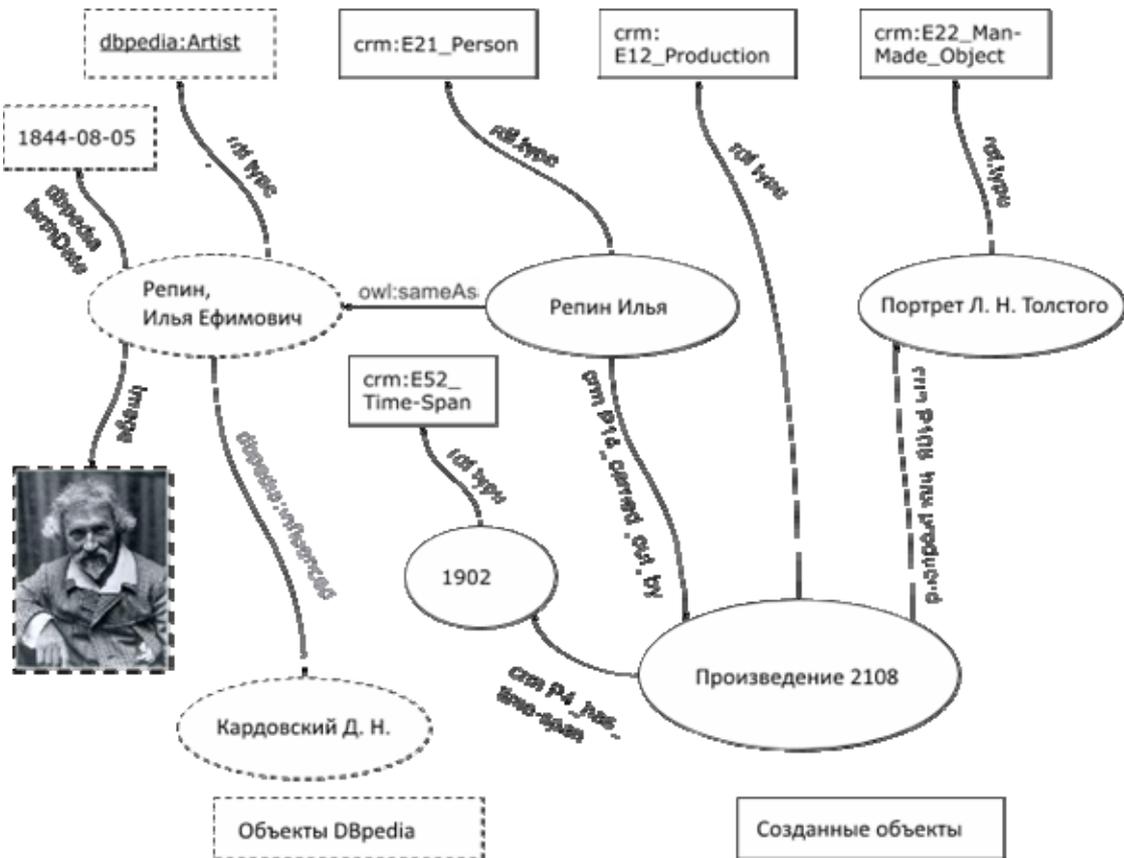


Рис. 2

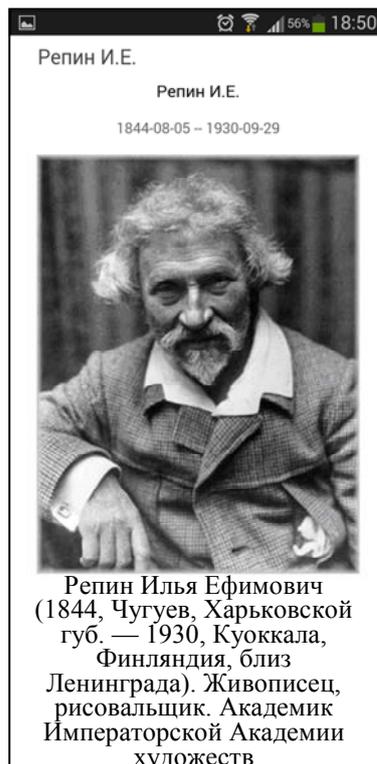


Рис. 3

Заключение. Предложенный подход к разработке мобильных клиентов с использованием предметно-ориентированной архитектуры, основанный на генерации модели предметной области исходя из внешнего описания, позволяет разработчику дополнительно формализовать части конечного приложения. Этап работы со знаниями в данном случае доступен не только программисту, так как отсутствует ограничение на использование неспециализированных инструментов (таких как языки программирования) для описания знаний. Эксперт предметной области может влиять на конечную модель, не только выступая в роли консультанта, но и непосредственно участвуя в создании формальной модели. Онтологическое представление модели упрощает реализацию предметно-ориентированной разработки за счет единого языка описания модели и разграничения контекстов предметной области.

СПИСОК ЛИТЕРАТУРЫ

1. Meng-Yen Hsieh, Hua-Yi Lin, Ching-Hung Yeh, Kuan-Ching Li, Bo-Shiung Wu. A Mobile Application Framework for Rapid Integration of Ubiquitous Web Services // Proc. of the 9th Intern. Conf. Ubiquitous Intelligence & Computing and Autonomic & Trusted Computing (UIC/ATC). 2012. P. 136—142.
2. Atkinson C., Kühne T. Model-driven development: a metamodeling foundation // IEEE Software. 2003. Vol. 20 (5), P. 36—41.
3. Agrawal A., Karsai G., Ledeczi A. An end-to-end domain-driven software development framework // Companion of the 18th Annual OOPSLA 03' ACM SIGPLAN Conf. Proc. 2003. P. 8—15.
4. Brambilla M., Cabot J., Wimmer M. Model-driven software engineering in practice // Synthesis Lectures on Software Engineering. Morgan & Claypool, 2012.
5. Völter M., Stahl T., Bettin J., Haase A., Helsen S. Model-driven software development: technology, engineering, management. N. Y.: John Wiley & Sons, 2013.
6. McGuinness D., Van Harmelen F. OWL web ontology language overview // W3C Recommendation, 2004 [Электронный ресурс]: <<http://www.w3.org/TR/owl-features>>.

7. *Fiannaca A., La Rosa M., Rizzo R., Urso A., Gaglio S.* An ontology design methodology for knowledge-based systems with application to bioinformatics // Proc. of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). 2012. P. 85—91.
8. *Stevenson G., Dobson S.* Sapphire: Generating Java runtime artefacts from OWL ontologies // Conf. Proc.: Advanced Information Systems Engineering Workshops — CAiSE Intern. Workshops. 2011. P. 425—436.
9. CIDOC Conceptual Reference Model [Электронный ресурс]: <<http://www.cidoc-crm.org/>>.

Сведения об авторах

- Дмитрий Александрович Замула** — аспирант; Университет ИТМО; кафедра информатики и прикладной математики-1; E-mail: zamula.dmitry@gmail.com
- Дмитрий Ильич Муромцев** — канд. техн. наук, доцент; Университет ИТМО; кафедра информатики и прикладной математики-1; E-mail: d.muromtsev@gmail.com

Рекомендована кафедрой
информатики и прикладной математики-1

Поступила в редакцию
31.08.15 г.

Ссылка для цитирования: *Замула Д. А., Муромцев Д. И.* Автоматизированное проектирование мобильных приложений на основе онтологий и семантических данных // Изв. вузов. Приборостроение. 2015. Т. 58, № 11. С. 939—944.

**COMPUTER AIDED DESIGN OF MOBILE APPLICATIONS
ON THE BASE OF ONTOLOGIES AND SEMANTIC DATA**

D. A. Zamula, D. I. Muromtsev

*ITMO University, 197101, St. Petersburg, Russia
E-mail: zamula.dmitry@gmail.com*

A new approach to creation of functional and structural elements of application, forming the architectural level of the domain model, is proposed. The approach is based on the use of metadata of the main data source. Due to specific requirements to the data source, the developed solution is reported to be most relevant to the client part of the distributed client-server systems with an RDF triple store as the main database. In practice, most of such clients are realized as mobile applications.

Keywords: ontology, domain-driven design, mobile application, RDF, SPARQL.

Data on authors

- Dmitry A. Zamula** — Post-Graduate Student; ITMO University, Department of Computer Science and Applied Mathematics-1; E-mail: zamula.dmitry@gmail.com
- Dmitry I. Muromtsev** — PhD, Associate Professor; ITMO University, Department of Computer Science and Applied Mathematics-1; E-mail: d.muromtsev@gmail.com

For citation: *Zamula D. A., Muromtsev D. I.* Computer aided design of mobile applications on the base of ontologies and semantic data // *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroyeniye*. 2015. Vol. 58, N 11. P. 939—944 (in Russian).

DOI: 10.17586/0021-3454-2015-58-11-939-944