

ГРАФИЧЕСКИЙ DSL ДЛЯ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Артур Гудиев, студент 2 курса маг. кафедры системного программирования
СПбГУ, arturgudiev93@gmail.com,
Александра Гражевская, Aspose, sagrapro7@gmail.com

Аннотация

В связи с увеличением количества платформ, языков и методов, использующихся в разработке мобильных приложений, задача выработки общей технологии довольно актуальна. Графические предметно-ориентированные языки (DSL) облегчают разработку программ путем применения абстракций конкретной предметной области.

В данной работе предполагается разработать архитектурный шаблон мобильного приложения и создать на его основе графический DSL, позволяющий описывать основную структуру мобильного приложения в терминах контроллеров, состояний и переходов между ними. Кроме того, необходимо реализовать автоматическую кодогенерацию для платформы UbiqMobile.

Введение

На сегодняшний день представлено большое количество платформ, языков и методов, использующихся в разработке мобильных приложений. Существующие инструменты мобильной разработки довольно разрозненны, и задача выработки общей технологии приложений по-прежнему актуальна.

Существуют различные способы высокоуровневого описания мобильных приложений - архитектурные паттерны mvc, ras, microkernel и т.д. Все эти паттерны были заимствованы из других сфер разработки ПО, довольно активно применяются при создании мобильных приложений, но не вполне соответствуют их природе.[1] Мобильные приложения отличаются от настольных и веб приложений, имеют свою специфику. Мобильные приложения обычно используются для более коротких сессий, более сфокусированы на выполнении конкретных задач.

Использование подходящего архитектурного паттерна позволяет значительно увеличить эффективность разработки приложения, но еще

большого результата можно достигнуть благодаря использованию графических языков. DSL – язык программирования в терминах конкретной предметной области, применяющийся для решения конкретного круга задач. Графические DSL языки помогают представлять приложения с помощью визуальных диаграмм, по которым затем генерируется код.

Целью данной работы является разработка архитектурного шаблона мобильного приложения и создания на его основе графического DSL, позволяющего описывать основную логическую структуру приложения в терминах контроллеров состояний и условий переходов между ними. По описанию приложения должен генерироваться его программный код в виде архитектуры разработанной модели. Разрабатываемый DSL должен иметь возможность надстройки над разными системами мобильной разработки.

Инструменты

Для генерации графического DSL языка используется технология **Modeling SDK for Visual Studio**. Modeling SDK - это платформа, предназначенная для разработки визуальных предметно-ориентированных языков. Modeling SDK позволяет интегрировать в Visual Studio собственные визуальные предметно-ориентированные языки программирования.[2] Разработка визуального предметно-ориентированного языка происходит следующим образом. Создается проект разработки DSL, затем разрабатывается и редактируется метамодель (описывается множество всех синтаксически корректных диаграмм), генерируются реализованные классы, затем происходит компиляция и отладка DSL-пакета в новом экземпляре Visual Studio.

Для программирования метамодели используется графический редактор Modeling SDK, но также можно переопределить или добавить новые методы в сгенерированные частичные классы языка C#. Для генерации используется язык описания шаблонов T4[3], представляющий из себя совокупность управляющих команд на языках C# или Visual Basic. В новом решении автоматически создаются проекты Dsl и DslPackage. В Dsl хранятся различные артефакты метамодели создаваемого DSL, а в DslPackage - настройки пользовательского интерфейса целевого графического редактора. Для хранения моделей используются доменные классы — абстракции сущностей хранимых данных в модели. Также есть классы фигур, свойства которых связаны со свойствами классов моделей посредством механизма событий в языке C#[4]. В SDK имеются способы представления и хранения, а также валидации моделей. Любой доменный класс можно пометить атрибутом, который наделяет его возможностью быть валидированным[5].

Описание графического DSL

Модель контроллеров и состояний была апробирована на мобильных приложениях разных классов и доказала свою эффективность. Но еще лучших результатов можно достигнуть, взяв эту модель за основу графического DSL для мобильных приложений.

Основными элементами языка являются **контроллер** и его **состояния**. Состояния размещаются на контроллере, могут соединяться между собой, а также с **портами** своего контроллера для описания условий входа и выхода из него.

Каждое состояние раскрывается в отдельную диаграмму, на которой описываются условия входа, выхода из состояния и его внутренняя логика. В логику состояний входит отображение экранных **форм**, обработка их событий, обращение к **сервисам**, источникам данных, проверка условий и т.д.

Часто при переходе в состояние происходит отображение экранной формы, а события её элементов могут обрабатываться в этом состоянии обращениями к сервисам, базе, или же сменой состояния. Для того, чтобы связать существующую экранную форму с состоянием, либо описать новую, используется элемент ShowForm.

Кодогенерация

Для генерации кода используется язык текста шаблонов T4. Основными компонентами языка являются директивы, блоки текста и блоки управления. Для генерации неизменяемого кода используются блоки текста, а динамические части реализуются с помощью блоков управления. Целевой платформой является платформа UbiqMobile.[6] В результате генерации получаются классы контроллеров. У каждого контроллера есть несколько состояний, представленных в виде типа перечисления. Процесс работы реализован в виде конечного автомата. По связям, заданным на диаграмме, реализуются шаблоны переходов в контроллерах. У контроллеров могут быть и порты. Они используются для переходов между контроллерами.

Пример мобильного приложения

Данное приложение позволяет войти в систему и получить код, который затем может быть использован. (см. Рис. 1) В приложении есть два

контроллера: LoginController и MainController. Присутствует также переключение между контроллерами, реализованное с помощью портов. В LoginController есть только одно состояние. В MainController присутствуют два состояния: состояние выбора опции и состояние, в котором волонтер может получить необходимый код.

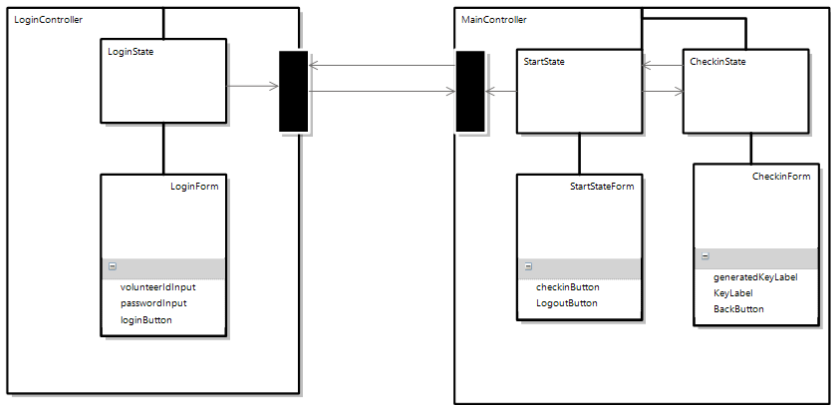


Рисунок 1: Схема приложения с авторизацией

Ниже представлены экранные формы UbiqMobile, соответствующие состояниям приложения. (см. Рис. 2)

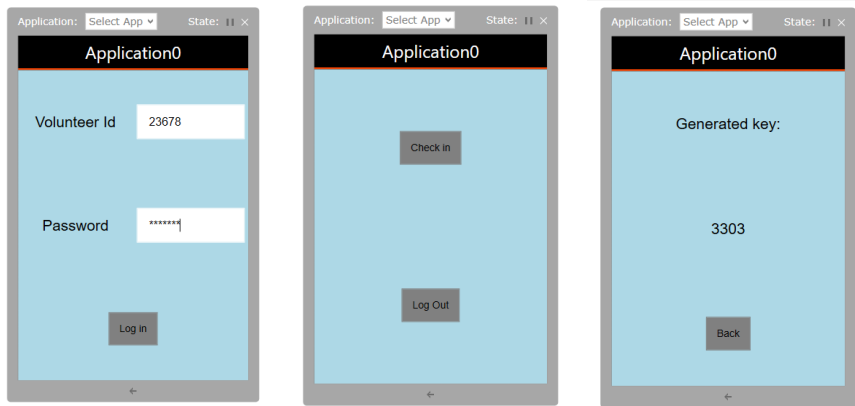


Рисунок 2: Экранные формы приложения

Заключение

В рамках данной работы были достигнуты следующие результаты:

- Реализован графический DSL для разработки мобильных приложений
- Добавлена возможность генерации кода для платформы UbiqMobile
- Реализованы демонстрационные примеры

Список литературы

[1] Applying MVC and PAC patterns in mobile Applications. Plakalovi'c D., Simi'c D.

[2] Modeling SDK for Visual Studio - Domain-Specific Languages
[https://docs.microsoft.com/ru-ru/visualstudio/modeling/ modeling-sdk-for-visual-studio-domain-specific-languages](https://docs.microsoft.com/ru-ru/visualstudio/modeling/modeling-sdk-for-visual-studio-domain-specific-languages)

[3] Code Generation and T4 Text Templates, [https://msdn.microsoft.com/ en-us/en-en/library/bb126445.aspx](https://msdn.microsoft.com/en-us/en-en/library/bb126445.aspx)

[4] Steve Cook Gareth Jones Stuart Kent Alan Wills. Domainspecific development with visual studio dsl tools. — Addison-Wesley Professional c 2007

[5] Никита Шигаров. Среда визуального программирования роботов на .NET, <http://se.math.spbu.ru/SE/diploma/2017/bmo/444-Shigarov-report.pdf>

[6] А.Н. Терехов, В.В. Оносовский. Платформа для разработки мобильных приложений UbiqMobile,

URL:http://www.math.spbu.ru/user/ant/all_articles/076_Terekhov_Onos_PlatformaUbiq.pdf