УДК 378.04

ВИЗУАЛЬНАЯ СРЕДА ПРОГРАММИРОВАНИЯ РОБОТОВ НА .NET

Д.А. Савочка

ЧОУ ВО ЮУ (ИУБИП)

4 год обучения, e-mail: dima_savochka@mail.ru

Научный руководитель

В.В. Храмов

к.т.н., профессор кафедры «Информационные технологии и прикладная математика» Южного университета (ИУБИП)

Аннотация: Представлено изложение основных возможностей программирования роботов на платформе .NET. Реализована визуальная среда программирования роботов и произведена апробация решения путем тестирования на реальном роботе
Ключевые слова:TRIK Studio, Modeling SDK, программирование, роботы

VISUAL ENVIRONMENT FOR PROGRAMMING ROBOTS ON .NET

D.A.Savochka

Scientifc adviser

V.V. Khramov

Abstract: The main features of robot programming on the .NET platform are presented. Implemented a visual environment for programming robots and Approbation of the solution was carried out by testing on a real robot
Keywords: TRIK Studio, Modeling SDK, programming, robots

Предметно-ориентированные языки (Domain SpecificLanguage или DSL) очень важны в современном мире программирования. Они служат удобным средством разработки архитектуры, бизнес-процессов и т. д. Также существуют визуальные предметно ориентированные языки, которые в некоторых важных случаях оказываются удобнее текстовых. Например, они полезны для обучения детей составлению алгоритмов, так как в данном случае изучение синтаксиса языка может быть затруднительно. На кафедре

ИТ и ПМ ИУБиП создан проект QReal [1] — кроссплатформенный свободно распространяемый под лицензией Apache License 2.01 инструмент с открытым исходным кодом, предназначенный для создания специализированных сред визуального программирования. Этот инструмент используется в ядре среды визуального программирования роботов TRIK Studio, которая позволяет составлять визуальные схемы алгоритмов, генерировать по ним код в несколько языков программирования, симулировать поведение роботов, а также отправлять программу на роботы Lego NXT, Lego EV3 или ТРИК [1]. TRIK Studio разработана на языке C++ с использованием библиотеки Qt [8].

Недавно стартовал новый проект REAL.NET [2], являющийся идейным продолжателем QReal и реализуемый на платформе .NET. В связи с этим возник интерес изучения технологии Modeling SDK, которая позволяет создавать собственные визуальные языки как плагины к Visual Studio [6], как аналога разрабатываемого REAL.NET с целью выяснить его сильные и слабые стороны, а также написать код, потенциально переиспользуемый в будущей реализации среды программирования роботов в REAL.NET.

Целью данной работы является разработка прототипа среды визуального программирования роботов ТРИК на .NET с помощью Modeling SDK, генерирующей код на JavaScript и использующей визуальный язык, близкий к языку среды TRIK Studio. Для достижениия этой цели были сформулированы следующие задачи:

- выполнить обзор возможностей среды программирования роботов TRIK Studio;
- исследовать структуру Modeling SDK;
- создать архитектуру решения;
- реализовать решение;
- выполнить апробацию.

TRIK Studio представляет из себя визуальную среду программирования роботов Lego Mindstorms NXT, Lego Mindstorms EV3 и ТРИК, которая

позволяет генерировать код на языках C#, F#, JavaScript, Pascal.ABC в случае конструктора ТРИК, NXT OSEK C в случае Lego NXT и байткод виртуальной машины в случае Lego EV3 по графическим диаграммам [7].

Визуальный язык среды TRIK Studio [3,4] устроен следующим образом. Различные управляющие блоки соединены направленными связями переходов, задающих порядок выполнения инструкций. У блоков и связей могут быть заданы свои параметры. Программа на визуальном языке может быть либо отлажена на компьютере (с опциональной посылкой пакетов на реального робота), либо сгенерирована в текстовый код и загружена на робота. В первом случае пользователь может создать виртуальный двухмерный мир, состоящий из стенок, цветных элементов и размеченных регионов и сэмулировать поведение робота в данной среде. Во втором случае визуальная программа генерируется в текстовый файл и отображается в текстовом редакторе, основанном на библиотеке qscintilla , а затем отправляется на робота.

Modeling SDK for Visual Studio является платформой по разработке визуальных предметно-ориентированных языков, которая позволяет интегрировать в Visual Studio свои визуальные предметноориентированные языки программирования. До Visual Studio 2017 предоставлялась как подключаемый модуль, начиная с 2017 версии интегрирована в среду [6].

Для программирования метамодели используется графический редактор Modeling SDK, но также можно переопределить или добавить новые методы в сгенерированные частичные классы языка C# вручную.

Для генерации используется язык описания шаблонов T4 [3], который представляет из себя совокупность управляющих команд на языках C# или Visual Basic, выделямых символами “<#” и “#>”, и текстовых блоков.

Шаблон преобразуется в программу на соответствующем языке, результатом работы которой будет код целевой системы.

В новом решении автоматически создаются проект Dsl и DslPackage [5]. Первый предназначен для хранения различных артефактов метамодели

создаваемого DSL, второй хранит настройки пользовательского интерфейса целевого графического редактора. В обоих проектах можно добавлять расширения классов на языке C# для расширения базовой функциональности нового редактора.

Для хранения моделей используются доменные классы — абстракции сущностей хранимых данных в модели. Так же есть классы фигур, свойства которых связаны со свойствами классов моделей посредством механизма событий в языке C# [9]. Эти классы представляют собой абстрации элементов диаграммы и их свойства видны пользователю в текстовых полях блоков на диаграмме. Можно задавать отношения между классами через создание специальных классов, тем самым обеспечивая задание правил для построения визуальных схем.

Способы представления и хранения моделей уже реализованы в SDK стандартными средствами языка C#.

Механизмы валидации моделей также реализованы в SDK. Любой доменный класс можно пометить атрибутом, который наделяет класс возможностью быть валидированным. После этого, нужно создать метод класса со своим специальным атрибутом, который указывает, при каких случаях будет запускаться этот метод — по вызову из меню, при сохранении или открытии модели, или из пользовательского кода. После выявления ошибок корректности свойств соответствующей модели, если такие имеются, нужно записать их в логгер ошибок соответствующего объекта ValidationContext, который хранит информацию о текущем процессе валидации.

Для решения поставленных задач была разработана архитектура решения. Редактор должен позволять строить как корректные, так и некорректные диаграммы на языке, приближенном к визуальному языку среды TRIK Studio. Никаких ограничений на структуру диаграммы не налагается в целях удобства пользователя.

Проверку корректности должен выполнять валидатор, запускаемый автоматически перед генерацией кода, либо по желанию пользователя.

Генератор кода в классе Generator должен генерировать код, который впоследствие либо будет записан в файл с последующим открытием в редакторе Visual Studio, либо отправлен на робота по каналу TCP/IP. Метод генерации запускается из класса шаблона RuntimeTextTemplate с передачей текущего экземпляра этого класса для вызова метода сохранения сгенерированной информации в этом объекте.

В ходе данной работы получены следующие результаты:

- выполнен обзор возможностей среды программирования роботов TRIK Studio;
- выполненобзор Modeling SDK for Visual Studio;
- разработана архитектура решения;
- реализована визуальная среда программирования роботов;
- произведена апробация решения путем тестирования на реальном роботе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. A. Kuzenkova A. Deripaska T. Bryksin Y. Litvinov V. Polyakov. QReal DSM Platform: An Environment for Creation of Specific Visual IDEs. —Proceedings of 8th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2013), SCITEPRESS, 2013, pp. 251-257.
2. Cheda K. Greenwood J. Bischof B. Pro Visual Studio .NET. — Apress; Softcover reprint of the original 1st ed. edition (August 30, 2004).
3. Code Generationand T4 Text Templates. — https://msdn.microsoft.com/en-us/en-en/library/ bb126445.aspx(датаобращения: 21.12.2017г).
4. Customizing the Isolated Shell. — https://msdn.microsoft.com/enus/en-en/library/ee 390883 .aspx (датаобращения: 21.12.2017г).
5. David B. Makofske Michael J. Donahoo Kenneth L. Calvert. TCP/IP Sockets in C#: Practical Guide for Programmers. — Morgan Kaufmann.
6. Modeling SDK for Visual Studio - Domain-Specific Languages. — https://msdn.microsoft.com/en-us/library/bb126259.aspx (датаобращения: 21.12.2017г).
7. Prieur Jean-Marc. [DSL Tools] Support of Nested shapes in Visual Studio 2008 SP1. —https://blogs.msdn.microsoft.com/jmprieur/2008/09/03/dsl-toolssupport-of-nested-shapes-in-visual-studio-2008-sp1/ (датаобращения: 21.12.2017г).
8. Qt Documentation. —http://doc.qt.io/ (дата обращения: 21.05.2017г).
9. Richter Jeffrey. CLR via C#: (4th Edition). — Microsoft Press; 4 edition, P. 249.