

Санкт-Петербургский государственный университет

Кафедра системного программирования

Кузьмина Елизавета Владимировна

Создание прототипа графического
редактора для DSM-платформы
REAL.NET

Курсовая работа

Научный руководитель:
к. т. н., доцент Литвинов Ю.В.

Санкт-Петербург
2017

Оглавление

Введение	3
1. Обзор	5
1.1. Существующие аналоги	5
1.2. Используемые технологии	5
2. Архитектура	9
3. Реализация	11
4. Заключение	13
Список литературы	14

Введение

Для решения специфических задач иногда бывает легче генерировать новые языки программирования, а не использовать уже имеющиеся. Получаемые языки позволяют создавать выразительные модели в терминах предметной области и могут быть использованы не только программистами, но и специалистами в данной области для решения поставленных перед ними задач.

При этом применение визуальных языков упрощает как понимание программы, так и работу с ней. Существуют исследования, которые говорят о том, что использование предметно-ориентированных визуальных языков во много раз повышает производительность труда программистов [1].

Возникает задача — быстрая генерация предметно-ориентированных графических языков. Для ее решения существуют специальные инструменты, называемые DSM-платформами¹. DSM-платформы помогают быстро создавать не только сами визуальные языки программирования, но также и инструменты для работы с ними, такие как репозитории, редакторы модели, средства проверки ограничений и генераторы кода.

Существует множество DSM-платформ, таких как Eclipse Graphical Modeling Project [2], Microsoft Modeling SDK [3], MetaEdit+ [4] и QReal [5]. Платформа QReal в течение нескольких лет разрабатывается на кафедре системного программирования нашего университета и базируется на многолетнем опыте коллектива кафедры в разработке графических языков. QReal — сложная система, ее трудно модифицировать с технической точки зрения, а любые изменения в системе могут сказаться на функционировании созданного на ее основе приложения для робототехники и помешать работе тысяч пользователей, поэтому целесообразно проводить исследования с использованием данной платформы.

¹Domain-Specific Modeling

Несмотря на результаты исследований и обилие существующих средств предметно-ориентированного визуального моделирования, такой подход к программированию применяется достаточно редко, что говорит о неудобстве имеющихся решений и подталкивает к созданию новых легковесных, удобных в использовании и простых в модификации инструментов.

Платформа REAL.NET задумывалась как набор конфигурируемых и переиспользуемых компонентов для создания визуальных предметно-ориентированных языков программирования.

Целью данной работы являлось создание графического редактора новой среды REAL.NET, для достижения которой необходимо было решить следующие задачи:

- обзор существующих средств для представления и отрисовки графов;
- разработка архитектуры редактора;
- реализация редактора.

1. Обзор

1.1. Существующие аналоги

QReal

QReal — это DSM-платформа, с помощью которой можно разрабатывать больше и сложные визуальные технологии, но которая при этом проста в использовании и позволяет быстро создавать визуальные языки программирования. С помощью системы QReal была создана среда работы с роботами, которая применяется для обучения программированию школьников.

В платформе QReal было реализовано “ядро” редактора, которое отвечает за общую функциональность всех редакторов разрабатываемых языков. На основе отличительных особенности каждого такого языка, которые можно узнать по его метамодели, создается подключаемый к “ядру” плагин этого языка. Такая архитектура позволяет добиться модульности системы и обеспечивает переиспользование компонентов, ведь можно просто внести изменения в “ядро” или добавить плагин и воспользоваться новыми возможностями во всех других редакторах, которые созданы на базе данной платформы.

Кроме того, в QReal реализовано так называемое “матамоделирование на лету” — подход, который позволяет уточнять метамодель непосредственно во время рисования модели. То есть совсем не обязательно использовать метаредактор для создания нового графического языка, достаточно просто по ходу рисования создавать необходимые элементы на палитре.

1.2. Используемые технологии

REAL.NET

Новая среда предметно-ориентированного визуального моделирования REAL.NET разрабатывается на платформе .NET и реализует принцип “глубокого метамоделирования”. Платформа позволит проводить

исследования в области предметно-ориентированного визуального моделирования, а также может быть встроена в любое .NET приложение.

На рисунке 1 представлена архитектура REAL.NET. Она включает в себя репозиторий, в котором хранится реализованная метамодель, редакторы и средство проверки ограничений, которые работают с этой метамodelью.

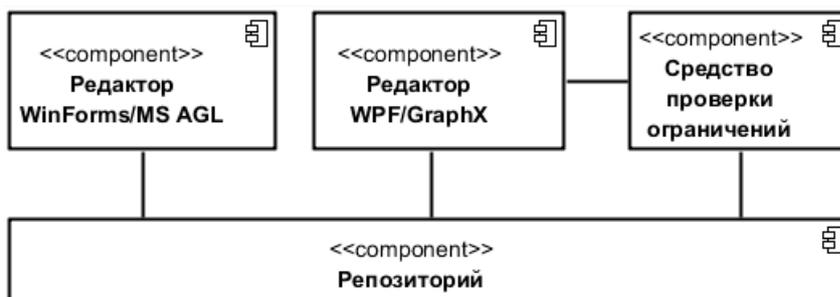


Рис. 1: Общая структура REAL.NET

Windows Forms

Windows Forms — это технология, разработанная компанией Microsoft для платформы .NET. Она представляет собой набор библиотек, позволяющих создавать приложения с полнофункциональным графическим интерфейсом. Данная технология является кроссплатформенной за счет того, что она поддерживается в Mono Framework - реализации .NET-машины и библиотек для операционных систем Linux и MacOS.

WPF

Платформа Windows Presentation Foundation, также разработанная компанией Microsoft для .NET Framework, позволяет создавать клиентские приложения с привлекательным пользовательским интерфейсом. WPF предоставляет модель программирования, содержащую декларативную (язык разметки XAML) и императивную составляющие. В отличие от устаревшей технологии Windows Forms, WPF включает новую модель построения пользовательских приложений (в основе WPF лежит мощная инфраструктура, основанная на DirectX), но она не является кроссплатформенной, а может работать только на операционной

системе Windows. Платформа WPF может естественно поддерживать возможности, необходимые дизайнерам.

QuickGraph

Данная библиотека предоставляет удобный интерфейс для представления графа, но требует сторонних библиотек для его визуализации, таких как GraphViz и MSAGL. Библиотека GraphViz представляет графы в удобном для чтения формате .dot, который потом подает на вход программе, включенной в пакет утилит и формирующей граф в виде графического, векторного или текстового файла. Нам же такой формат изображения не подходит, так как нужно уметь редактировать модель прямо на сцене. MSAGL — удобный инструмент, который поддерживает отрисовку с масштабированием элементов и различными видами узлов и связей, но он больше не поддерживается в QuickGraph.

MSAGL

Microsoft Automatic Graph Layout — набор библиотечных инструментов для визуализации и редактирования графов, разработанный компанией Microsoft. Как было отмечено ранее, MSAGL — удобная библиотека для работы с графами. Но у нее есть недостаток - отсутствие документации. С помощью данной библиотеки и технологии Windows Forms реализован один из редакторов системы REAL.NET, который позволяет системе быть кроссплатформенной.

GraphX

GraphX — это библиотека с открытым исходным кодом, которая позволяет визуализировать графы, предоставляя для этого множество инструментов, а также поддерживает различные алгоритмы компоновки графов. Данная библиотека умеет сразу представлять граф в графической форме без отсылки к другим компонентам. Она может быть использована как вместе с WPF, так и с Windows Forms. Неудобство при работе с GraphX доставляет неполная документация. В библиотеке GraphX граф представляется с помощью класса GraphExample.

В этот граф входят вершины и ребра, представленные соответственно классами `DataVertex` и `DataEdge`. Для отображения модели применяется класс `GraphArea`, у которого есть свойство `LogicCore`. Это свойство определяет алгоритм компоновки графа и содержит свой граф уже рассмотренного класса `GraphExample`, элементы которого отображаются на `GraphArea` при помощи классов `VertexControl` и `EdgeControl`.

2. Архитектура

В результате анализа существующих средств для реализации редактора, описываемого в данной работе, была выбрана платформа WPF как современная среда для разработки приложений с богатой моделью рисования, а также библиотека GraphX как удобная библиотека с множеством возможностей для рисования графов, которая может быть использована с WPF.

Требованиями к редактору были обеспечение переиспользуемости его компонентов и модульности, чтобы как и в система QReal можно было получить новые решения путем дополнения и параметризации кода самой системы.

В ходе решения поставленной задачи была разработана архитектура редактора, реализующая паттерн программирования Model-View-Controller, который помогает разделить данные приложения, пользовательский интерфейс и управляющую логику на отдельные компоненты. Класс Controller отвечает за связь отделенных друг от друга модели и окна взаимодействия с пользователем. Получившаяся архитектура показана на рисунке 2.

Пользователь совершает действия в главном окне программы, класс которого отвечает только за визуализацию. Вызывается контроллер, который при помощи класса Model инициирует изменения в модели, лежащей в репозитории. После того как в репозитории были совершены необходимые действия, модель генерирует событие, на которое подписан класс Graph. В этом классе лежит представление графа в терминах библиотеки GraphX. С помощью этого события Graph узнает о необходимых изменениях, добавляет соответствующие элементы и генерирует следующее событие, на которое подписано главное окно. Оно же в свою очередь добавляет на сцену редактора элементы управления, соответствующие появившимся в графе и в репозитории элементам.

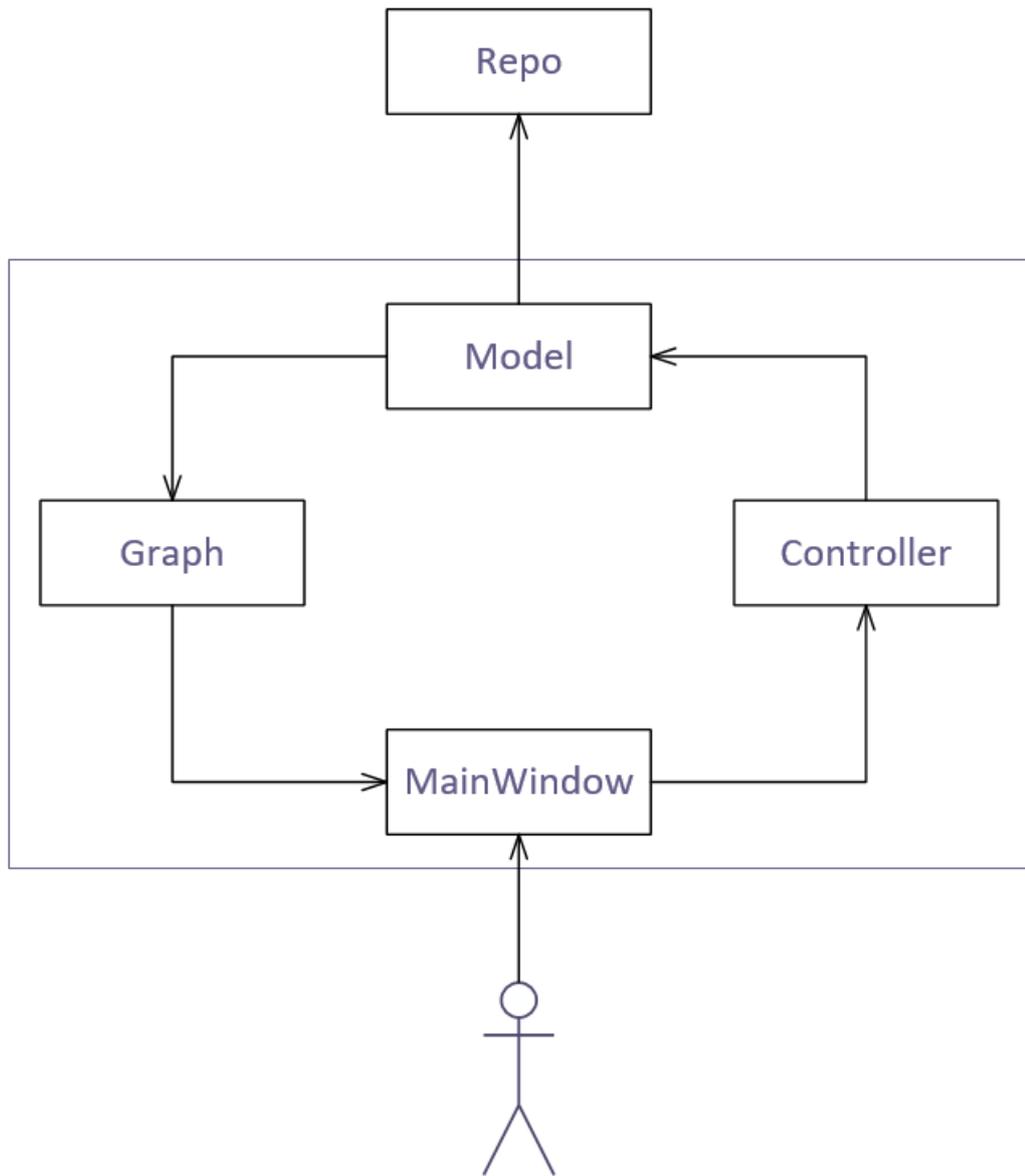


Рис. 2: Архитектура редактора

3. Реализация

При реализации редактора использованы такие возможности библиотеки GraphX, как редактирование моделей непосредственно на сцене, а также поддержка различных видов узлов и ребер. В настоящее время вид вершин и ребер пользователь может определить в тексте программы. Всем типам узлов и связей, которые были объявлены в репозитории, он присваивает фигуры для их отображения. После этого пользователь оперирует данными объектами для создания и редактирования модели из репозитория на сцене. На рисунке 3 показано, как выглядит созданный в рамках данной работы редактор системы REAL.NET.

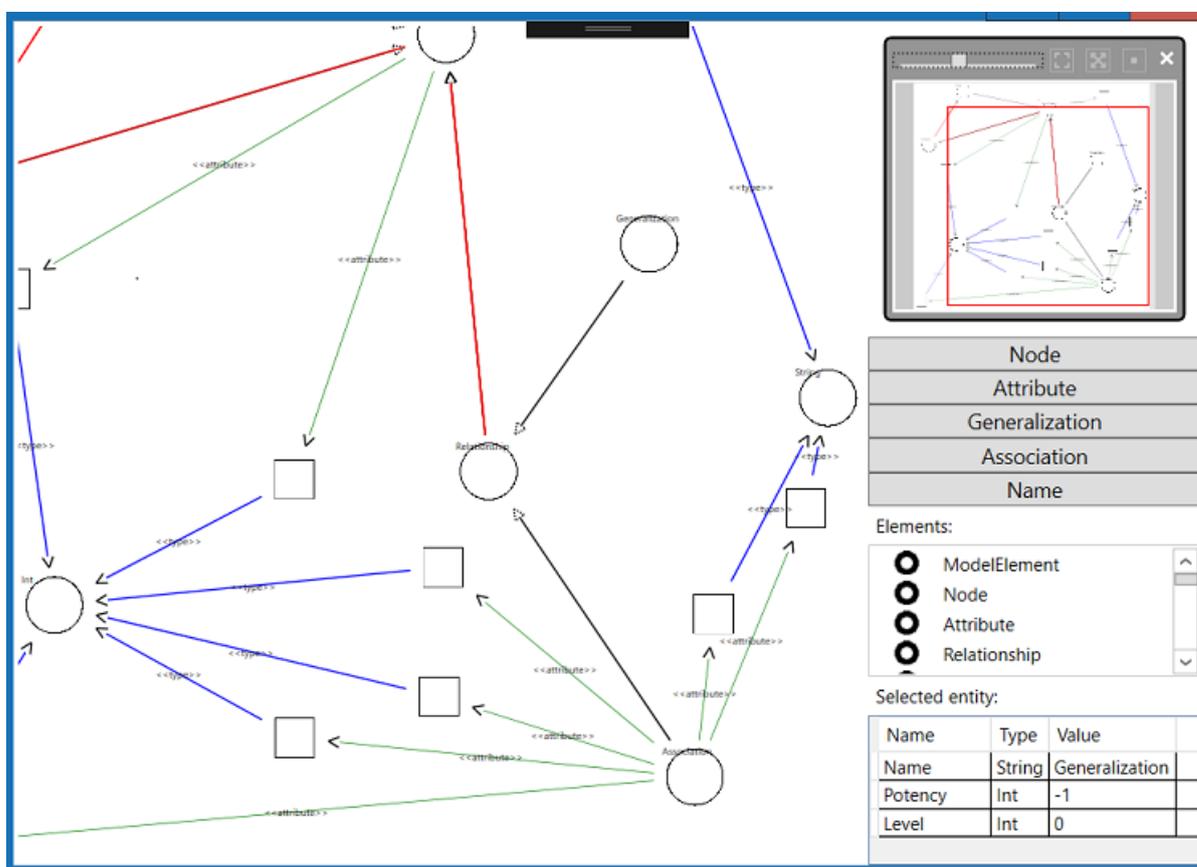


Рис. 3: Редактор системы REAL.NET

Редактор представляет собой WPF форму, которая включает в себя:

- сцену для создания и редактирования графа;

- миникарту сцены, с помощью которой можно масштабировать изображение модели;
- палитру со всеми видами узлов и связей, представленных в репозитории;
- список всех элементов модели;
- редактор свойств модели.

Для удобства работы с графом рядом с его элементами можно отобразить их свойства в виде меток. Единственный случай, когда эта функция не поддерживается — случай петель. В такой ситуации вершина, инцидентная ребру указанного вида, отображается на графе со специальным значком, который информирует о наличии у нее какого-то количества инцидентных петель, сами ребра при этом не отрисовываются, соответственно не показываются и их свойства.

Чтобы удобно расположить граф на сцене, пользователю предоставляется возможность перемещать вершины, а также изменять форму связей, делая из прямых ребер кривые, огибающие несвязанные с ними элементы модели.

4. Заключение

В рамках данной курсовой работы были получены следующие результаты:

- сделан обзор существующих средств для работы с графами на платформе .NET;
- разработана архитектура редактора;
- реализован редактор.

Список литературы

- [1] Kelly S. Tolvanen J.-P. Visual domain-specific modeling Benefits and experiences of using metaCASE tools // International Workshop on Model Engineering, at ECOOP.— 2000.— URL: http://dsmforum.org/papers/Visual_domain-specific_modelling.pdf (online; accessed: 24.05.2017).
- [2] R.Gronback. EclipseModelingProject:ADomain-SpecificLanguage(DSL) Toolkit. — 2009. — P. 736.
- [3] S. Cook G. Jones S. Kent A.C. Wills. Domain-specific development with Visual Studio DSL Tools. — 2007. — P. 576.
- [4] S. Kelly J.-P. Tolvanen. Domain-specific modeling: enabling full code generation. — 2008. — P. 444.
- [5] Wiki страница QReal.— URL: <https://github.com/qreal/qreal/wiki> (online; accessed: 24.05.2017).