

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Зайнуллин Егор Евгеньевич

Редактор REAL.NET

Курсовая работа

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

Санкт-Петербург
2018

Оглавление

Введение	3
1. Обзор	4
1.1. Существующие решения	4
1.2. Что уже реализовано	6
2. Описание решения	7
2.1. Панель инструментов	7
2.2. Архитектура отмены операций	7
3. Апробация	9
Заключение	10
Список литературы	11

Введение

Область применения предметно-ориентированных визуальных языков широка. Она включает в себя робототехнику, промышленность, математические вычисления. При этом стоит отметить, такие языки интуитивно понятны и просты, а значит их использование достаточно эффективно. Существует некоторое количество инструментов для создания визуальных языков, например: *MetaEdit+* [2], *Eclipse MP* [3], а также язык, разрабатываемый и поддерживаемый кафедрой системного программирования СПбГУ *QReal* [5]. Несмотря на это, визуальные языки встречаются довольно редко. Отсюда можно сделать вывод: инструменты для их создания недостаточно удобны [4].

Поэтому нами было поставлена задача создать REAL.NET — аналог QReal, который бы позволял .NET разработчикам работать с визуальными языками.

На данный момент ведется работа по созданию редактора, работающего с графовыми предметно-ориентированными языками.

Постановка задачи

1. Реализовать панель инструментов, причем таким образом, чтобы кнопки можно было подключать динамически
2. Реализовать поддержку отмены операций, в частности, действий на сцене: удаления, добавления, переноса и т. д.

1. Обзор

1.1. Существующие решения

На данный момент существует множество аналогичных решений. Рассмотрим следующие из них.

MetaEdit+

MetaEdit+ – инструмент, позволяющий создавать диаграммы, матрицы и таблицы. Состоит из двух компонентов:

- MetaEdit Modeler — поддерживает работу с моделью
- MetaEdit Workbench — отвечает за создание графического языка и генерацию кода

Для построения метамодели используется язык GOPRR (graph, object, property, relation, role). После построения GOPRR модели ее можно открыть в Workbench и использовать для создания необходимой предметной модели. Метамодель может быть изменена ”на лету”, что позволяет не запускать заново приложение при изменении метамодели. Для работы с моделью используется интерпретативный подход.

Несмотря на огромную функциональность MetaEdit+, работать с ним не всем представляется возможным, так как это коммерческий продукт. При этом отсутствует какая-либо возможность добавления сторонних расширений.

Eclipse Modeling Project

Eclipse Modeling Project (EMP) — набор инструментов для создания и визуализации моделей, для их сравнения, для генерации кода на различных языках программирования. Ядром EMP является проект EMF, его цель - создание метамodelей для новых графических языков. Метамodelи разрабатываются с помощью специального языка Ecore. Элементу Ecore-модели сопоставляется Java класс. Сохранение

этих элементов происходит с помощью сериализации. Модель может задаваться многими способами, такими как: XML описание, UML, Java код и т. д. ЕМР включает в себя множество других проектов, которые в совокупности решают много больше задач, чем просто создание визуальных языков.

Проект активно развивается до сих пор.

QReal

QReal — инструмент для быстрого создания визуальных языков, разрабатываемый СПбГУ. Расширение функциональности редактора осуществляется посредством добавления новых плагинов. При разработке нового языка необходимо работать с метаредактором, с помощью которого происходит добавление и изменение метаязыка. Для каждого блока задается код для генерации. На основе QReal была создан TRIK Studio, визуальный язык для взаимодействия с роботами.

Из-за его привязки к TRIK Studio и огромного количества написанного кода кардинально менять и экспериментировать с QReal стало невозможно. REAL.NET призван заменить его в этом плане, а также дать возможность .NET разработчикам проектировать новые визуальные языки.

Microsoft Modeling SDK

Microsoft Modeling SDK [1] — средство Visual Studio, которое позволяет создавать несложные графические графовые языки. Работать с ними придется в Visual Studio.

Отсюда и основные недостатки: привязка к Visual Studio и C#, невозможность расширения. При этом стоит отметить, что фактическая поддержка этого проекта прекратилась Microsoft.

Вывод

REAL.NET пробует другой подход, чем указанные выше средства. Он не делает различие между моделью и метамоделью. Это решает

несколько проблем, например: не надо иметь отдельный редактор для модели и метамодели, не надо думать, является ли элемент частью метамодели или нет. Такой подход называется "глубокое метамоделирование". Наш проект берет наработки QReal, например, расширение функциональности будет осуществляться посредством добавления плагинов, а также учитывает ошибки своего "предка".

1.2. Что уже реализовано

Редактор REAL.NET использует библиотеку GraphX для отрисовки графа. Она была выбрана по причине ее большой функциональности и красивого отображения графов на сцене.

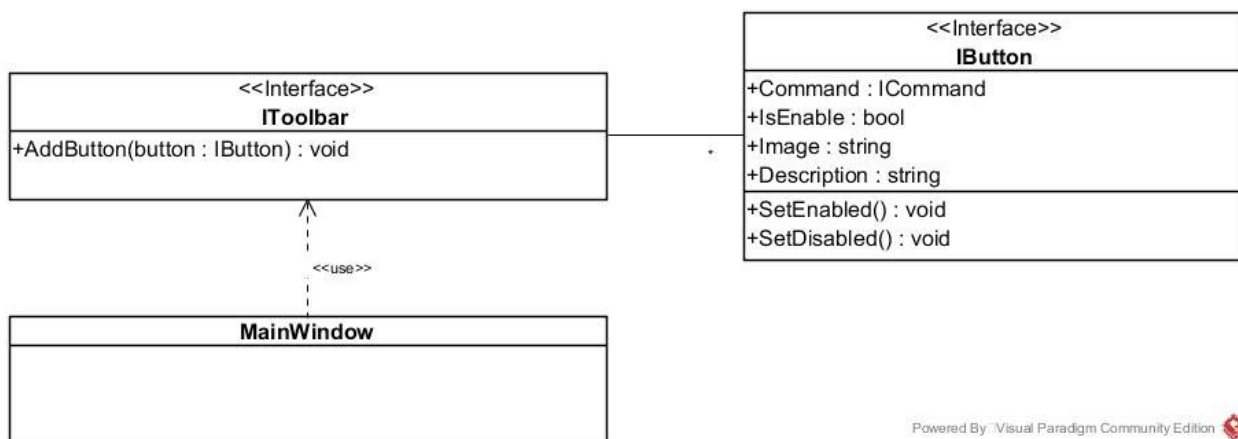
- Реализована возможность добавления, удаления вершин, добавления ребер
- Регистрация элемента, выбранного на палитре

2. Описание решения

2.1. Панель инструментов

Панель инструментов позволяет динамически добавлять кнопки, то есть элементы, реализующие интерфейс `IButton` (рис. 1). В XAML прописывается привязка к коллекции кнопок, нажатие на которые вызывает действие, завернутое в команду, которая, в свою очередь, содержится в интерфейсе кнопки. Для того, чтобы команда выполнялась, пришлось создать "адаптер", который бы приводил ее к интерфейсу `Windows.Input.Command`.

Рис. 1: Архитектура панели



Powered By Visual Paradigm Community Edition

2.2. Архитектура отмены операций

При выполнении какого-либо действия на сцене вызывается метод `SceneActionsRegister`, он заворачивает выполненное действие и его отмену в `ICommand` и добавляет в стек (рис. 2).

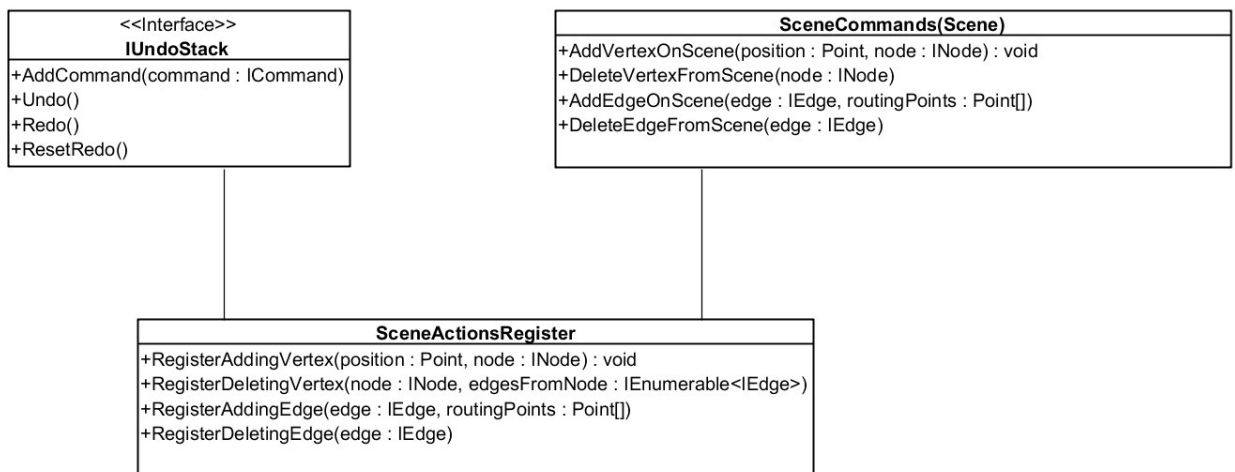
На данный момент поддерживаются отмена следующих операций

- Добавления вершины: `SceneActionsRegister` запоминает позицию только что созданного элемента, создает два делегата, один из которых добавляет элемент заново с сохраненной позицией, другой — удаляет его; объединяет их в одну команду и сохраняет в стек

- Удаления вершины: SceneActionsRegister действует аналогичным образом, только делегаты создаются в обратном порядке.
- Добавление ребра: SceneActionsRegister запоминает точки перегиба ребра, далее создает два делегата аналогичным образом с добавлением вершины.
- Удаление ребра: SceneActionsRegister действует аналогично добавлению.

В дальнейшем планируется добавить поддержку отмены переноса вершин и изменения точек перегиба ребер.

Рис. 2: Архитектура отмены операций на сцене



3. Апробация

- Для тестирования панели инструментов были добавлены несколько кнопок, которые при нажатии выводили надпись на консоль приложения.
- Чтобы проверить корректную работу отмены операций на сцене, нами проводилось несколько тестовых сценариев
 1. Удаляли изначальные элементы на сцене, потом отменяли эту операцию, потом заново ее повторяли
 2. Добавляли новые элементы на сцену, после перетаскивали, затем их удаляли, отменяли это действие
 3. Меняли точки перегиба у вершин, после их удаляли и отменяли это действие

Все тесты были выполнены успешно, программа выдавала ожидаемый результат.

Заключение

1. Была реализована панель инструментов, которая позволяет динамически добавлять кнопки
2. Реализована поддержка отмены операций, а также действий на сцене: удаления и добавления ребер и вершин

Список литературы

- [1] S. Cook G. Jones S. Kent A.C. Wills. Domain-specific development with Visual Studio DSL Tools. — Crawfordsville, Indiana, USA : Addison-Wesley, 2007. — 576 p.
- [2] S. Kelly J.-P. Tolvanen. Domain-specific modeling: enabling full code generation. — Hoboken, New Jersey, USA : Wiley-IEEE Computer Society Press, 2008. — 444 p.
- [3] А. Сорокин Д. Кознов. Обзор проекта Eclipse Modeling Project. Системное программирование. — Изд. СПбГУ, 2010. — С. 6–31.
- [4] Литвинов Ю.В. Кузьмина Е.В Небогатиков И.Ю Алымова Д.А. Среда предметно-ориентированного программирования REAL.NET. — GitHub. — 2017. — URL: <https://github.com/yurii-litvinov/articles/blob/master/2017-realNet/realNet.pdf> (дата обращения: 04.06.2018).
- [5] Терехов А.Н. Брыксин Т.А. Литвинов Ю.В. QReal: платформа визуального предметно-ориентированного моделирования. — 2013. — Т. 6 из Программная инженерия. — С. 11–19.