Средства задания правил генерации в QReal

Выполнила: Дерипаска Анна, 545 группа
Научный руководитель : ст. преп. Литвинов Ю.В.
Рецензент: ст. преп. Луцив Д.В.

Санкт-Петербург, 2014

Введение

- Визуальное моделирование
- DSM-подход
  - для конкретной задачи
- CASE-системы
  - создание программ на визуальном языке
- metaCASE-системы
  - создание новых визуальных языков
  - дополнительные возможности
  - генерация кода по диаграмме

Постановка задачи

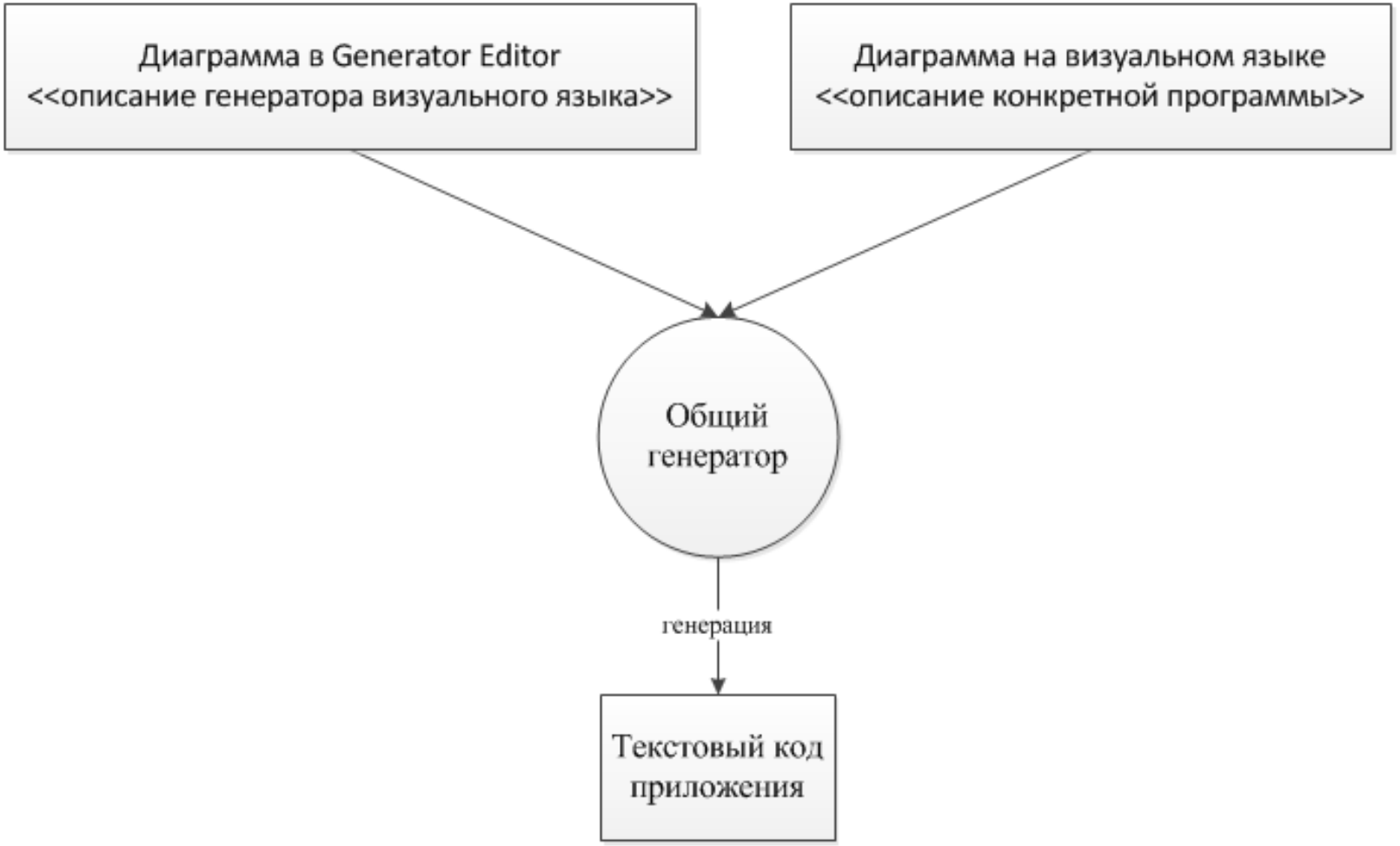
- Цель:
  - поддержать возможность задания правил генерации в среде QReal
- Задачи:
  - сделать обзор существующих решений данной задачи в других подобных системах
  - разработать метод задания правил генерации (визуальный язык)
  - разработать метагенератор
  - апробировать решение на существующих в QReal визуальных языках

Обзор

- QReal
  - metaCASE-система
- Существующие решения
  - Microsoft Visualization and Modeling SDK
  - Actifsource (Eclipse)
  - MetaCase MetaEdit+
  - Текстовый язык Geny (в QReal)
- Используемые решения
  - Генератор для QReal:Robots

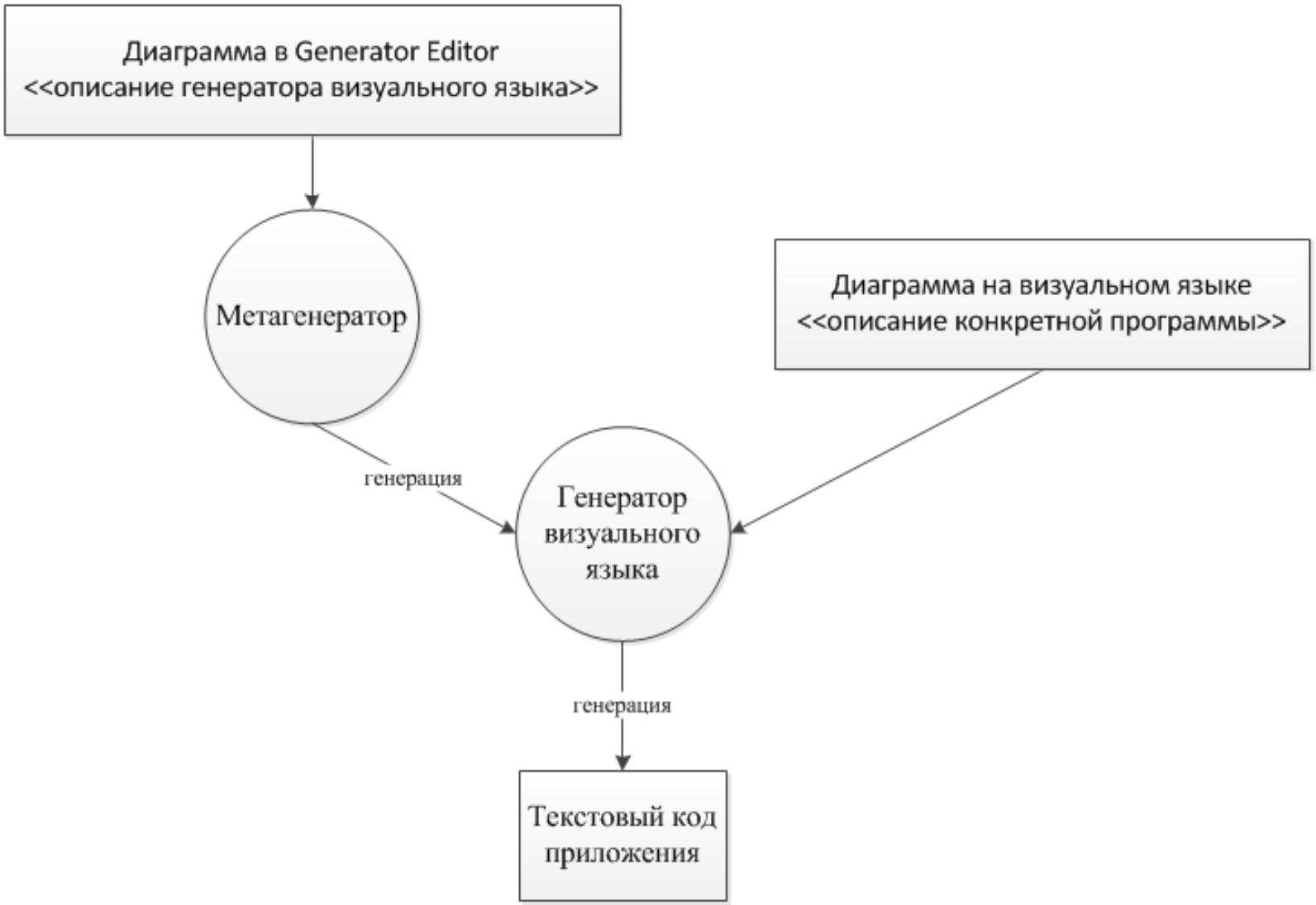
Архитектура (1)

Первый вариант архитектуры:



Архитектура (2)

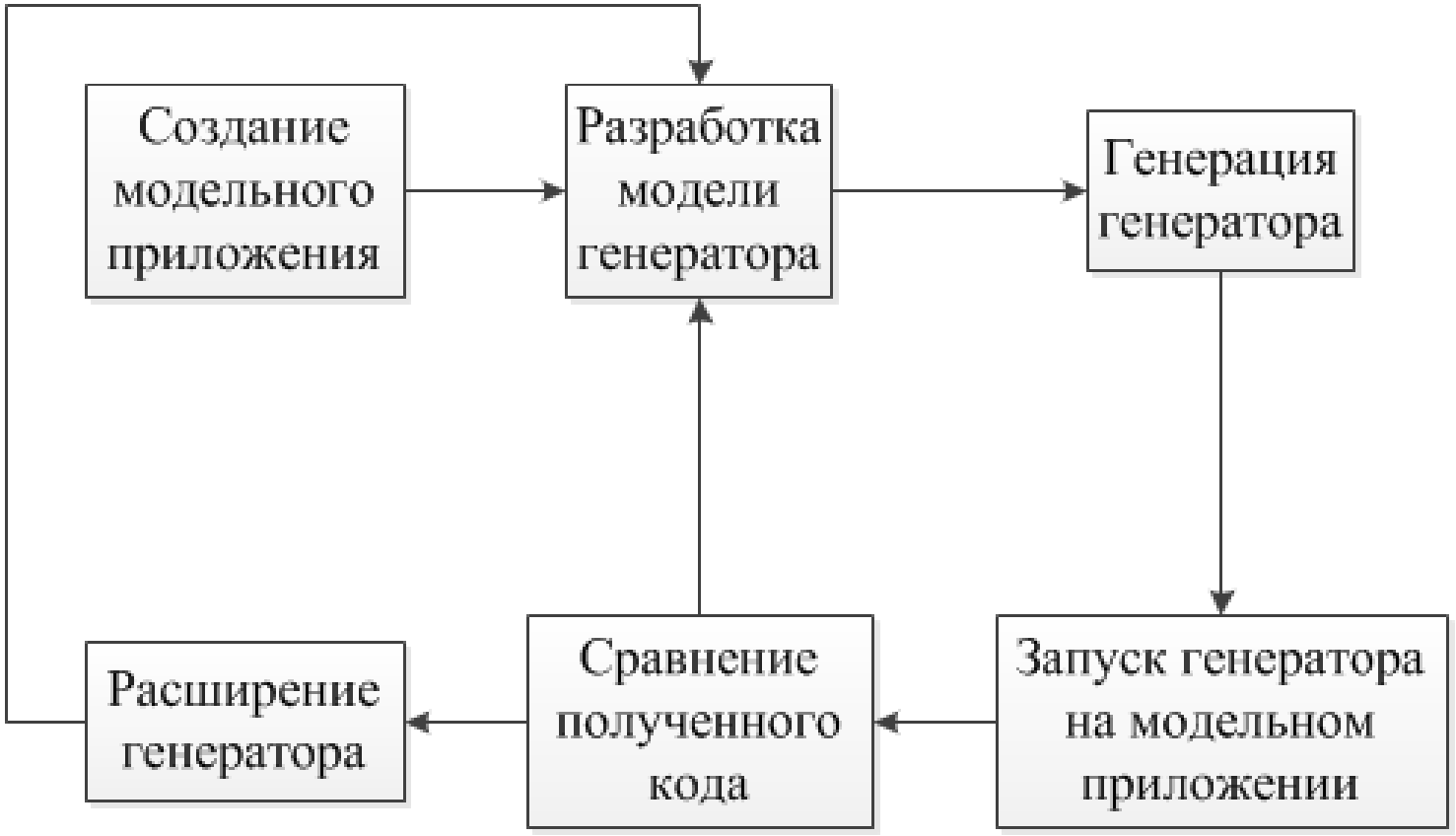
Второй вариант архитектуры (был выбран):



Визуальный язык

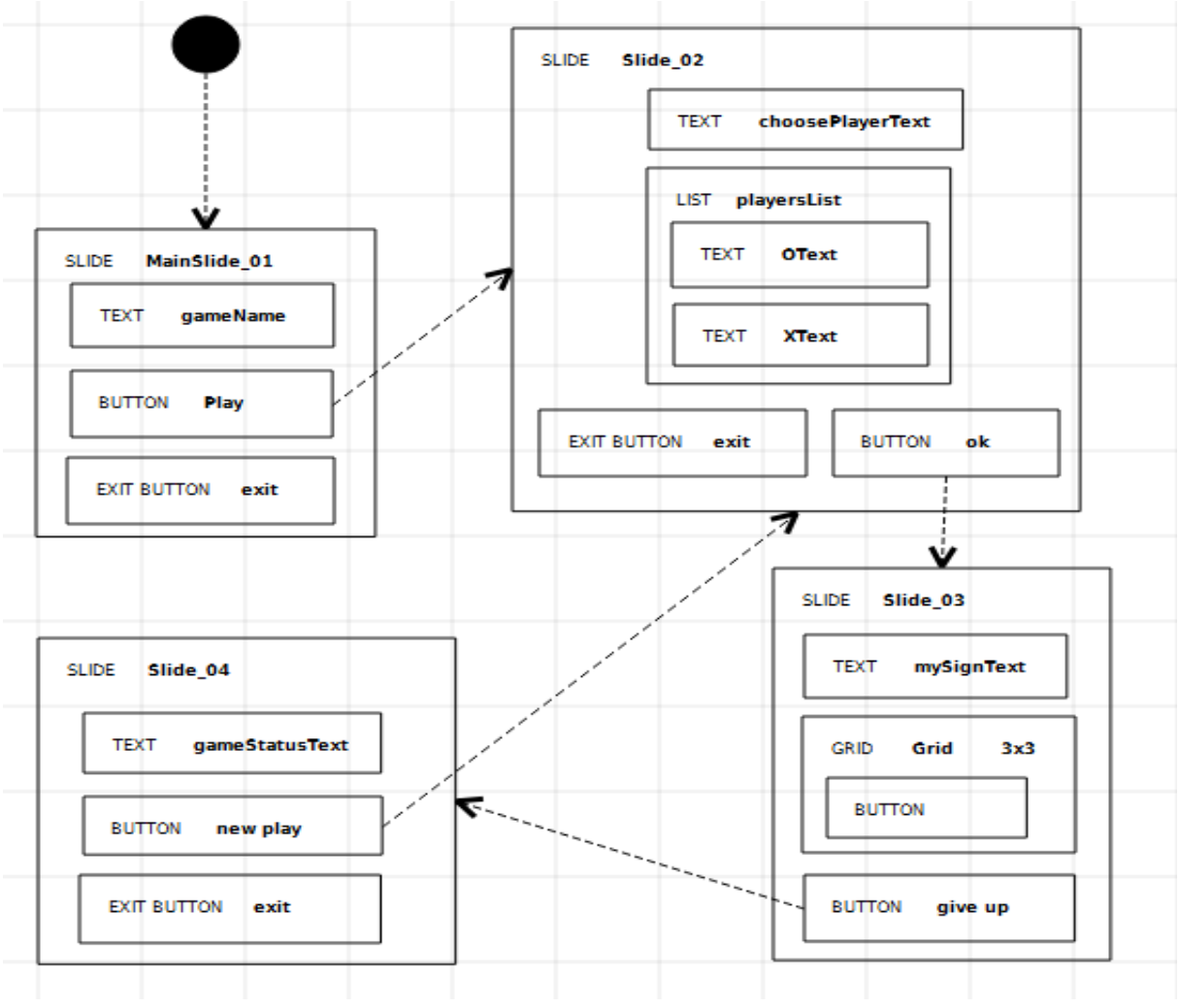
- Корневые элементы
  - Generator Root
  - Generator Diagram
- Базовые элементы
  - Application
  - Semantic
  - Semantic For Edge
  - Template
  - Foreach
- Элементы для задания конвертеров
  - Converter
  - Converter Semantic
  - Converter Foreach
  - Merge
  - Read
- Элементы-связи
  - To Edge
  - Control Flow

Процесс создания генератора



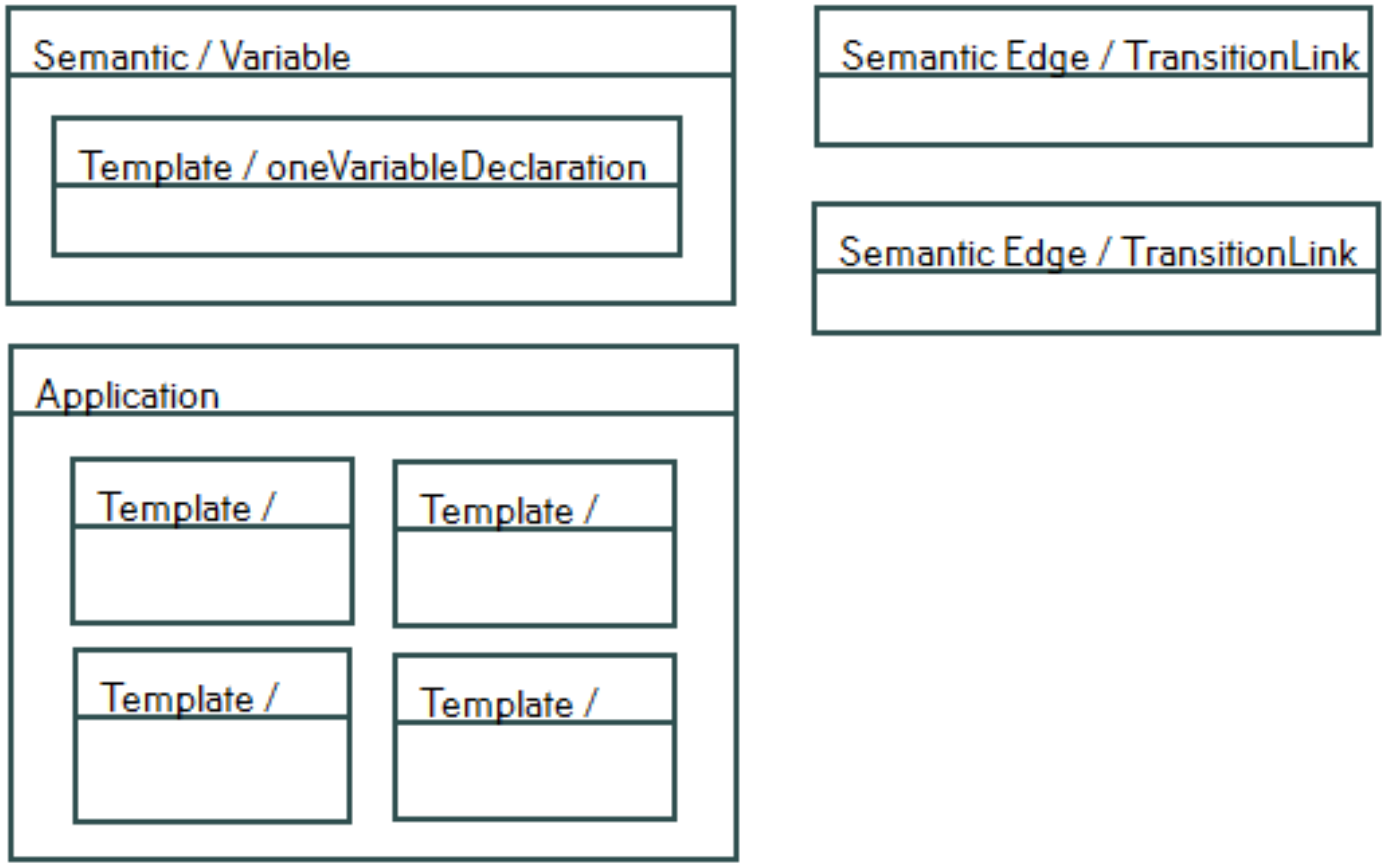
Апробация (1)
QReal:QUbiq, экранные формы

1). Диаграмма, описывающая онлайн-игру в крестики нолики на визуальном языке QUbiq Presentation Editor



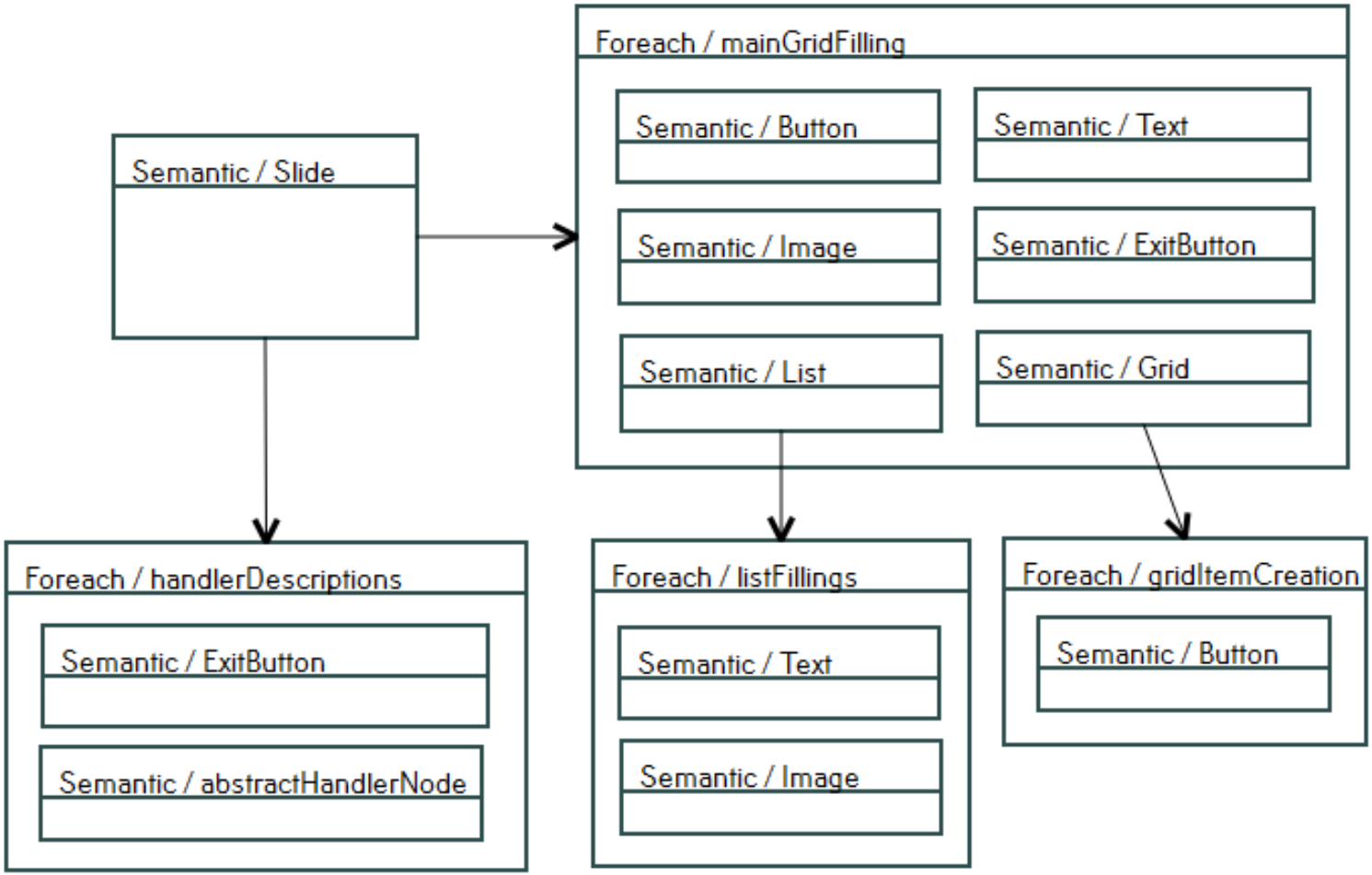
Апробация (2)
QReal:QUbiq, экранные формы

2). *Модель генератора [часть 1]*



Апробация (3)
QReal:QUbiq, экранные формы

2). *Модель генератора [часть 2]*



Апробация (4)
QReal:QUbiq, экранные формы

2). *Модель генератора [примеры шаблонов кода]*

GeneratorDiagram | Текстовый редактор

```
1  public static ##type## ##name## {get; set;}
2
```

GeneratorDiagram | Текстовый редактор

```
1  namespace @@programName@@
2  {
3      using System;
4      using Ubiq.Graphics;
5      using System.Collections.Generic;
6      ....
7      public static class @@programName@@Variables
8      {
9  @@elements_Variable@@
10     }
11 }
12
```

Апробация (5)
QReal:QUbiq, экранные формы

3). *Общая структура основной части полученного генератора:*

*foreach(*semanticEdge *in List<***SemanticForEdge***>)*

код, сгенерированный по свойствам элемента semanticEdge

*foreach(*templateNode *in List<***Template***>)*

код, сгенерированный по свойствам элемента templateNode

*foreach(*semanticNode *in List<***Semantic***>)*

код, сгенерированный по свойствам элемента semanticNode

*foreach(*foreachNode *in List<***Foreach***>)*

код, сгенерированный по свойствам элемента foreachNode

*foreach(*semanticNode *in List<***Semantic***>)*

<<аналогичная генерации элементов типа **Semantic**>>

код, сгенерированный для элемента типа **Application** :

*foreach(*templateNode *in List<***Template***>)*

код, сгенерированный по свойствам элемента templateNode

Апробация (6)
QReal:QUbiq, экранные формы

4). *Пример сгенерированного кода модельного приложения для MainSlide_01:*


```
private VisualElement CreateMainSlide_01()
{
        var mainGrid = new Grid(1, 2)
        {
                Margin = new Thickness(5, 5, 5, 5),
                VerticalAlignment = VerticalAlignment.Stretch,
                HorizontalAlignment = HorizontalAlignment.Stretch
        };
        var button_Button_01 = new Button();
        button_Button_01.Pressed += OnButton_01Clicked;
        button_Button_01.Pressed += ByButton_01Handler;
        mainGrid[1, 2] =  button_Button_01;

        var button_ExitButton_01 = new Button();
        button_ExitButton_01.Pressed += OnExitButton_01Clicked;
        button_ExitButton_01.Pressed += ByExitButton_011Handler;
        mainGrid[1, 3] =  button_ExitButton_01;

        var text_Text_01 = new TextBlock() { Text = "gameName" };
        mainGrid[1, 1] =  text_Text_01;

        return mainGrid;
}
```

Результаты

- Поддержана возможность задания правил генерации в среде программирования QReal
- Сделан обзор существующих решений данной задачи в других средах программирования
- Разработан язык задания правил генерации для некоторого визуального языка
- Реализован метагенератор для структурных языков
- Решение апробировано на существующем языке системы QReal (QUbiq, экранных форм)