

Компонента для визуализации данных с сохранением связи с исходным представлением

Курсовая работа Саравайского Михаила, 361 группа
Руководитель: Григорьев Семен



Проблема

В системном программировании, и, в частности, при реинжиниринге ПО возникает очень много информации для анализа

- Стеки вызовов
- Структуры данных
- Базы данных
- Грамматики

Такие данные удобно представлять в виде графов

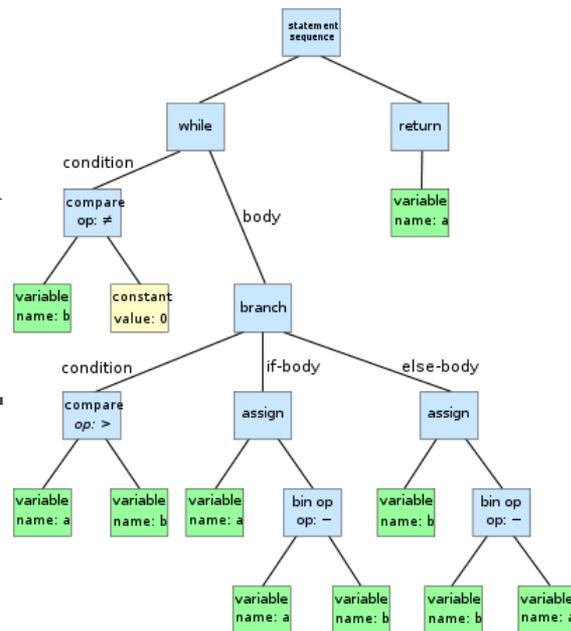


Цель работы

Основные цели:

Создать библиотеку, которая позволит визуализировать необходимые данные, сохраняя связь с исходными данными

```
1 Imports System.Data.SqlClient
2
3 Public Class Log
4
5     Private _cmd As New SqlCommand()
6
7     Private _objectType As String = ""
8     Private _objectId As String = ""
9     Private _counter As Integer = 0
10
11 Public Sub New(ByVal objectType As String, ByVal objectId As String)
12     _objectId = objectId
13     _objectType = objectType
14 End Sub
15
16 Public Sub Add(ByVal field As String, ByVal value As String)
17     If Not String.IsNullOrEmpty(_cmd.CommandText) Then _cmd.CommandText &= vbCrLf
18     _cmd.CommandText &= String.Format("INSERT INTO [log] ([object], [object_id], [field], [value], [date]) " & _
19         "VALUES (@obj, @obj_id, @field(0), @value(0), getDate());", _
20         _counter)
21     If Not _cmd.Parameters.Contains("@obj") Then _cmd.Parameters.Add("@obj", SqlDbType.VarChar).Value = _objectType
22     If Not _cmd.Parameters.Contains("@obj_id") Then _cmd.Parameters.Add("@obj_id", SqlDbType.VarChar).Value = _objectId
23     _cmd.Parameters.Add(String.Format("@field{0}", _counter), SqlDbType.VarChar).Value = field
24     _cmd.Parameters.Add(String.Format("@value{0}", _counter), SqlDbType.VarChar).Value = value
25     _counter += 1
26 End Sub
27
28 Public Sub Flush()
29     If String.IsNullOrEmpty(_cmd.CommandText) Then Return
30     Using myConn As New SqlConnection(_connectionString)
31         myConn.Open()
32         _cmd.Connection = myConn
33         _cmd.ExecuteNonQuery()
34     End Using
35 End Sub
36
37 End Class
```



Постановка задачи

От конечного решения требуется:

- визуализировать граф исходя из некоторых данных
- предоставлять информацию о действиях с графом в смысле изначального представления.
- обеспечить простоту использования как для разработчиков, так и для конечных пользователей

Данная библиотека должна быть разработана для .NET Framework и предоставлять возможность встраивания в WPF приложения

Обзор существующих решений

	Визуализация произвольных графов	Сохранение связи с исходным представлением	различные алгоритмы раскладки	Исходный код
MSAGL	+	+	-	-
yFiles	+	-	+	Высокая стоимость
GraphSHarp	+	-	+	+

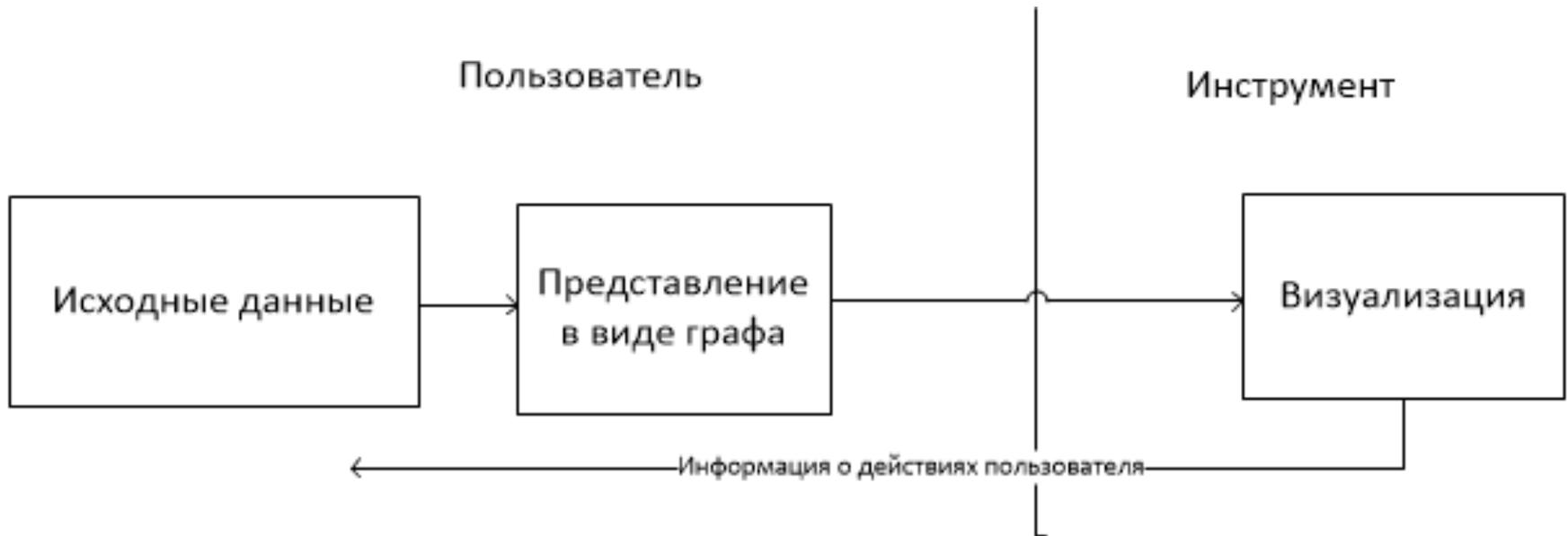
Особенности реализации

Отказ от автоматической генерации представления графа из произвольных данных.



Особенности реализации

Отказ от автоматической генерации представления графа из произвольных данных.



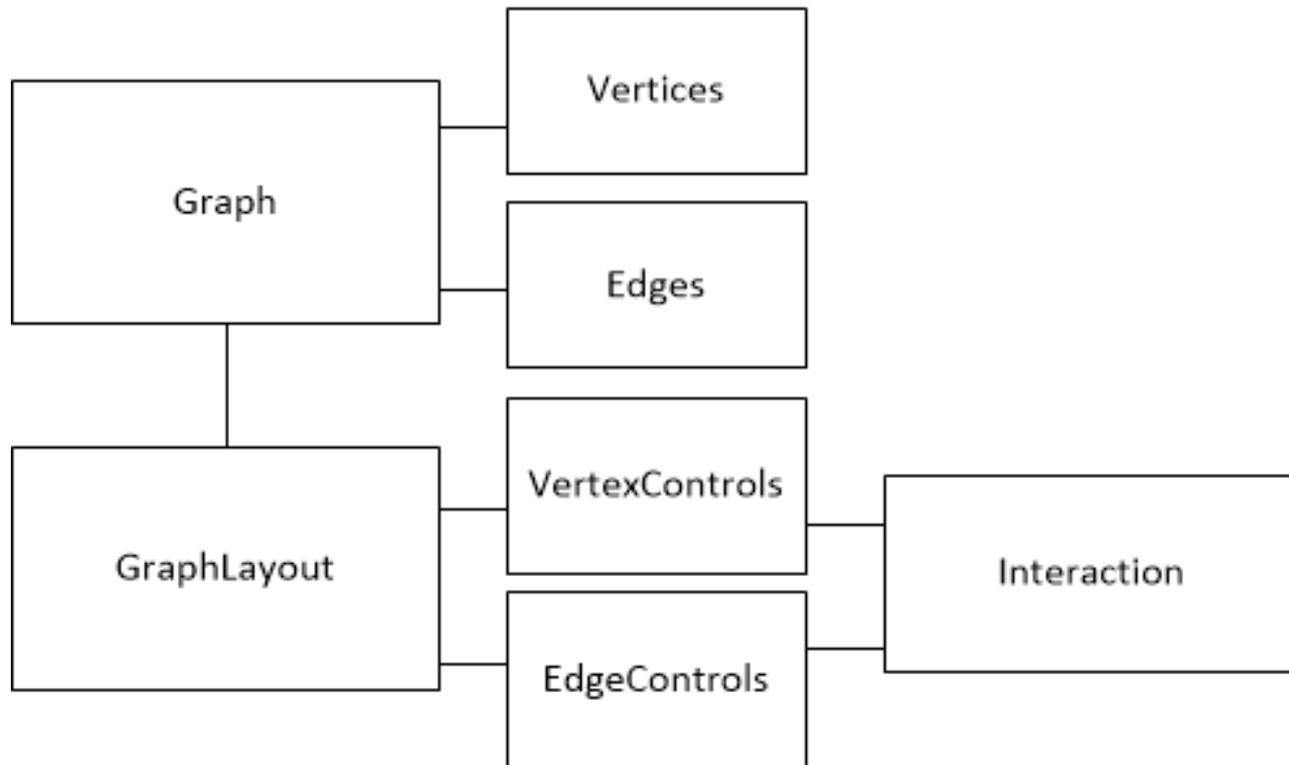
Особенности реализации

В ходе работы было реализовано:

- Создание QuickGraph оберток для данных
- Перехват информации о действиях пользователя
- Создание удобного механизма управления визуализацией

Особенности реализации

GraphSharp до модификации



Особенности реализации

GraphSharp после модификации

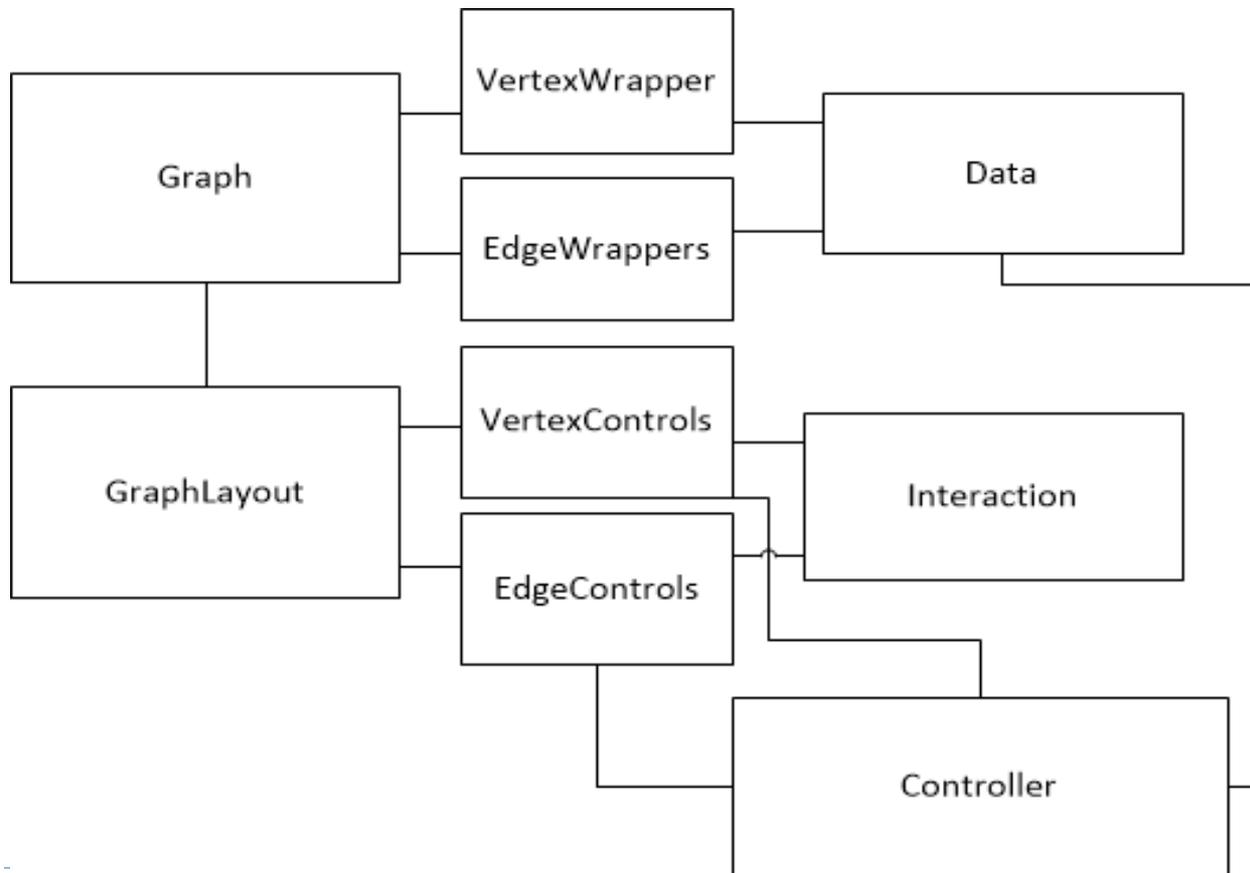


Схема работы



Результаты

Реализована библиотека со следующими возможностями:

- Визуализация данных
- Уведомление о действиях с графом
- Удобное для разработчика использование

- Работающий тест