Министерство образования и науки Российской Федерации
(МИНОБРНАУКИ РОССИИ)
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (ТГУ)
Факультет Информатики
Кафедра программной инженерии

КУРСОВАЯ РАБОТА

РАЗРАБОТКА ВИЗУАЛЬНОГО КОНСТРУКТОРА СЕТЕЙ ДЛЯ СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ПРОЦЕССОВ МАССОВОГО ОБСЛУЖИВАНИЯ

Чувилёв Константин Александрович

Руководитель
канд. тех. наук, доцент
_____ А.Н. Моисеев
подпись
«___» _____ 2013 г.

Студент группы № 1401
_____ К.А. Чувилёв
подпись

Томск 2013

Содержание

Введение

При моделировании различных процессов порой трудно обойтись без визуального представления. Наличие визуализации помогает лучше ориентироваться в моделируемой системе, а также предоставляет совсем иной подход к моделированию. Однако мало отображать процессы, пользователю необходимо иметь возможность полностью смоделировать систему, пользуясь наглядным визуальным редактором.

Целью данной работы является разработка визуального конструктора для системы имитационного моделирования процессов массового обслуживания ODIS [3] (более подробное описание этой системы представлено в п. 1). Для достижения поставленной цели требуется решить следующие задачи:

1. Проанализировать существующие решения на рынке.
2. Разработать визуальный конструктор сетей для системы имитационного моделирования процессов массового обслуживания.
3. Интегрировать визуальный конструктор сетей в систему имитационного моделирования процессов массового обслуживания ODIS.

В качестве языка программирования был выбран C# [4], так как система ODIS разработана с использованием этого же языка, вследствие чего затраты на интеграцию визуального конструктора в систему ODIS минимальны. Библиотека GMap.NET [5] (описание использования представлено в п. 3.3) выбрана из-за удобства использования, а также открытости исходного кода.

1. Краткое описание системы ODIS

1.1. Терминология теории массового обслуживания

Теория массового обслуживания (ТМО) используется для описания и исследования моделей процессов функционирования систем, которые мгновенно изменяют свое состояние под воздействием внешних и внутренних событий. К таким системам, в частности, относятся экономические системы, системы передачи информации по сетям связи и многие другие. Представление моделей таких процессов в терминах ТМО имеет свою специфику. В частности, объекты, подлежащие обработке внутри системы обслуживания, в русскоязычной литературе называют заявками. Такие объекты, обычно, поступают на вход системы массового обслуживания (СМО), затем некоторое время находятся внутри нее и в том или ином виде покидают систему. Именно эти объекты и формируют характеристики, подлежащие исследованию (состояние системы, длина очереди, выходящий поток и т.д.). Поскольку объекты-заявки должны поступить на вход системы, то обычно ведут речь о входящем потоке заявок.

Для представления механизмов обработки заявок в СМО используется понятие обслуживающего прибора (или блока обслуживающих приборов). Именно в нем производятся некоторые манипуляции над объектом-заявкой, которые называют обслуживанием. В результате этих действий заявка может быть обслужена и покинуть систему. Если же по каким-либо причинам приборы не могут обработать заявку, то она либо покидает систему необслуженной (такую ситуацию еще называют отказом), либо помещается в специальный накопитель (буфер), ожидая возможности быть обработанной.

Таким образом, можно выделить следующие основные классы элементов системы массового обслуживания, с помощью которых можно конструировать модели различного вида:

- заявка;
- источник входящих заявок – гипотетический объект, порождающий входящий поток заявок;
- блок обслуживающих приборов;
- буфер.

Сеть массового обслуживания (СеМО) представляет собой совокупность конечного числа обслуживающих узлов, в которой циркулируют заявки, переходящие в соответствии с

матрицей маршрутизаций из одного узла в другой. [2]

1.2. Используемые компоненты системы

ODIS [2] — это система имитационного моделирования процессов массового обслуживания. В рамках данной работы разработанный визуальный конструктор сетей массового обслуживания будет интегрирован в эту систему. Для этого нам понадобится компонент Matrix из комплекса программ ODIS, предназначенный для работы с матрицей маршрутизаций. Сеть массового обслуживания задаётся матрицей маршрутизаций, определяющей вероятности перехода заявок из одного узла в другой. Также для описания СеМО необходима матрица, задающая вероятности перехода заявок от входных потоков к блокам обслуживающих приборов.

2. Требования к системе

Функциональные требования:

1. Система должна визуализировать сеть массового обслуживания.
   1.1. Система должна отображать источники входящих заявок, блоки обслуживающих приборов и выходной буфер.
   1.2. Система должна отображать направления переходов потоков заявок между устройствами.
2. Система должна предоставлять возможность задания структуры и параметров сети массового обслуживания.
   2.1. Система должна предоставлять возможность добавления в сеть массового обслуживания новых входных буферов, блоков обслуживающих устройств, а также потоков заявок между ними.
   2.2. Система должна предоставлять возможность изменения вероятности переходов потоков заявок.
3. Система должна предоставлять возможность применения геопозиционирования структуры сети.
   3.1. Система должна привязывать объекты сети массового обслуживания к географическим координатам.
   3.2. Система должна объединять объекты сети, расположенные на малом расстоянии друг от друга, в логические группы.

Нефункциональные требования идентичны нефункциональным требованиям к комплексу программ ODIS [2], частью которого является разрабатываемая система.

3. Модель приложения

3.1. Общие принципы

В разрабатываемой системе для построения пользовательского интерфейса используется система визуальных слоёв, состоящая из: слоя работы с картами, слоя визуализации объектов сети массового обслуживания, а также из слоя элементов управления. На диаграмме (рис. 1) представлена схема организации слоёв, каждый из которых реализован следующими классами:

Рис. 1. Схема организации визуальных слоёв.

1. DrawCanvas — предназначен для манипуляции и визуализации элементов системы массового обслуживания.
2. GMapControl — предназначен для отображения и взаимодействия с картами местности.
3. DrawControl — предназначен для работы с работы с элементами управления, а также отвечает за работу остальных слоёв.

Самым ближним к пользователю является слой DrawControl, который получает различные команды от пользователя и определённым образом их выполняет. Некоторые команды, предназначенные для других слоёв, DrawControl рассылает ответственному за них слою, где происходит их обработка.

На диаграмме (рис. 2) представлена схема организации классов, предназначенных для работы с сетью массового обслуживания, состоящая из:

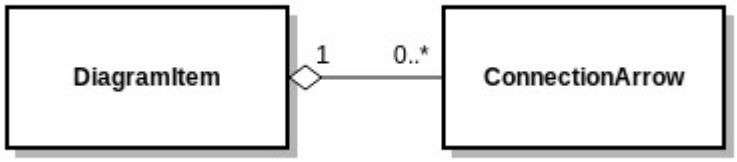

Рис. 2. Схема организации классов, предназначенных для работы с сетью массового обслуживания

1. DiagramItem — реализация обслуживающего устройства, входного и выходного буфера.
2. ConnectionArrow — реализация потока заявки.

3.2. Технические решения

При изменении масштаба схемы имелась необходимость объединять блоки в группы в зависимости от расстояния между блоками, а также убирать группировку, при приближении карты. Для решения данной проблемы использовался следующий алгоритм:

1. Введём переменную *minDistance* > 0, показывающее минимальное расстояние между блоками, при котором не стоит их объединять в группу.
2. Построим матрицу весов расстояний *A* между блоками. В результате получим неориентированный граф, размерность которого равна количеству блоков.
3. Преобразуем полученную матрицу следующим образом: если элемент *A[i,j]* > *minDistance*, тогда присваиваем *A[i,j]* = 0, иначе *A[i,j]* = 1. Таким образом мы получим матрицу смежности, определяющую блоки, находящиеся в непосредственной близости друг от друга.
4. С помощью алгоритма поиска в глубину выделим списки блоков, подлежащих группировке.

Теперь перейдём непосредственно к группировке. Нам важно, чтобы работа с групповым элементом не отличалась от работы с обычным элементом. Для реализации данного поведения используем типовое решение «**Компоновщик**» [1].

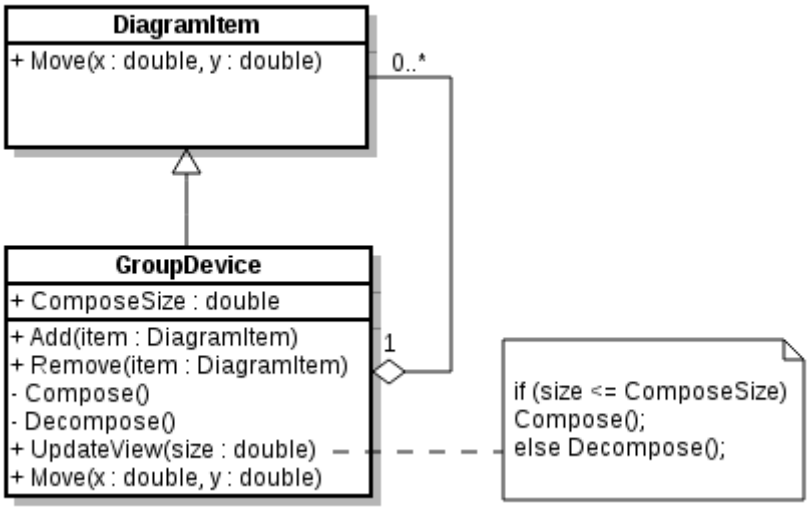

Рис. 3. Реализация типового решения «Компоновщик».

На диаграмме (рис. 3) представлена реализация данного решения. Методы Add и Remove ответственны за добавление и удаление блоков из группы. Методы Compose и Decompose необходимы для визуальной агрегации блоков в группу и, наоборот, для разбиения на составляющие. За выбор того, что сейчас необходимо делать (группировать или разбивать на множество блоков), ответственен метод UpdateView с входным параметром size (уровень приближения карты). Важно подчеркнуть необходимость переопределения метода Move в классе GroupDevice из-за необходимости симметричного перемещения сгруппированных элементов в соответствии с перемещением объекта GroupDevice.

Также дважды используется типовое решение «**Одиночка**» [1], предоставляя глобальные точки доступа к экземпляру карты (MapHelper (описание методов представлено в п. 3.3)) и к элементам сети массового обслуживания (DiagramItemManager).

3.3. Использование компонента GMap.NET

Gmap.NET [5] — это бесплатный элемент управления для платформы .NET с открытым исходным кодом. Он предоставляет удобные методы для работы с картами, предоставляя на выбор десятки карт, доступных бесплатно (в том числе Google Maps, Yandex Maps и другие). В рамках разрабатываемой системы данный элемент управления был использован в качестве дальнего слоя, над которым был надстроен слой для работы с элементами сети массового обслуживания. Для более удобной работы с данным элементом управления был создан специальный класс-помощник MapHelper (рис. 4), ответственный за глобальный доступ к картам из любой точки приложения, а также содержащий все

необходимые методы для работы с ними. Было уместно реализовать его с помощью решения «**Одиночка**» [1].

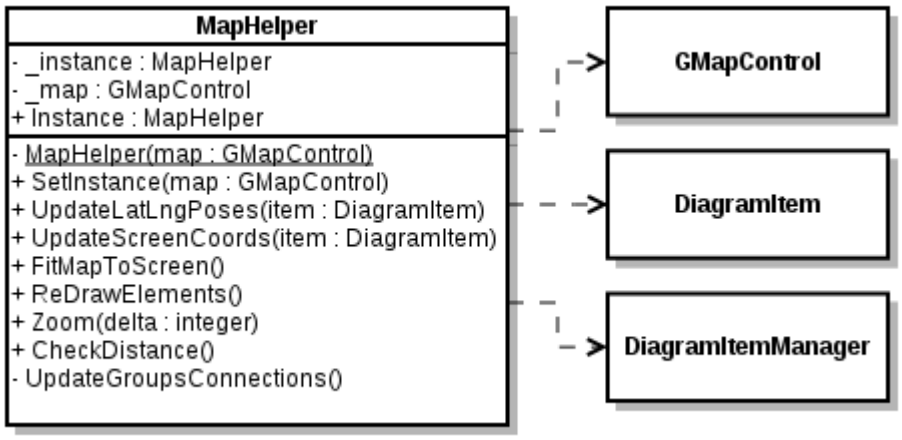

Рис. 4. Реализация класса MapHelper.

Ниже представлено более подробное описание основных методов класса MapHelper:

- UpdateLatLngPoses — метод принимает на вход ссылку на объект класса DiagramItem, для которого необходимо обновить географические координаты, в соответствии с текущим визуальным положением элемента.
- UpdateScreenCoords — метод принимает на вход ссылку на объект класса DiagramItem, для которого необходимо обновить визуальное положение на экране, в соответствии с его географическими координатами.
- FitMapToScreen — метод, вычисляющий на карте крайний левый верхний объект DiagramItem, а также крайний правый нижний объект DiagramItem. На основе координат этих элементов на карте строится прямоугольник, содержимое которого необходимо центрировать и увеличить до необходимого размера, чтобы содержимое полностью влезало в экран, тем самым мы увидим минимальное полное покрытие карты, в котором содержатся все наши элементы.
- ReDrawElements — метод, который обновляет визуальное положение элементов при взаимодействии пользователя с картой (перемещение области просмотра и изменение уровня приближения).
- Zoom — метод, ответственный за масштабирование карты, принимающий на вход уровень необходимого масштаба.
- CheckDistance — метод, ответственный за группировку близко расположенных

элементов DiagramItem (подробное описание алгоритма представлено в п. 3.2).

- UpdateGroupsConnections — метод, предназначенный для визуального представления потоков заявок между группами (на основании связей между элементами, входящих в эти группы). Вызов происходит в методе CheckDistance..

Класс MapHelper представляет из себя глобальную точку доступа к экземпляру карты (GmapControl). На диаграмме последовательности (рис. 5) представлено решение для проблемы объединения элементов в группы при изменении масштаба схемы (п. 3.2) с помощью применения класса-помощника MapHelper.

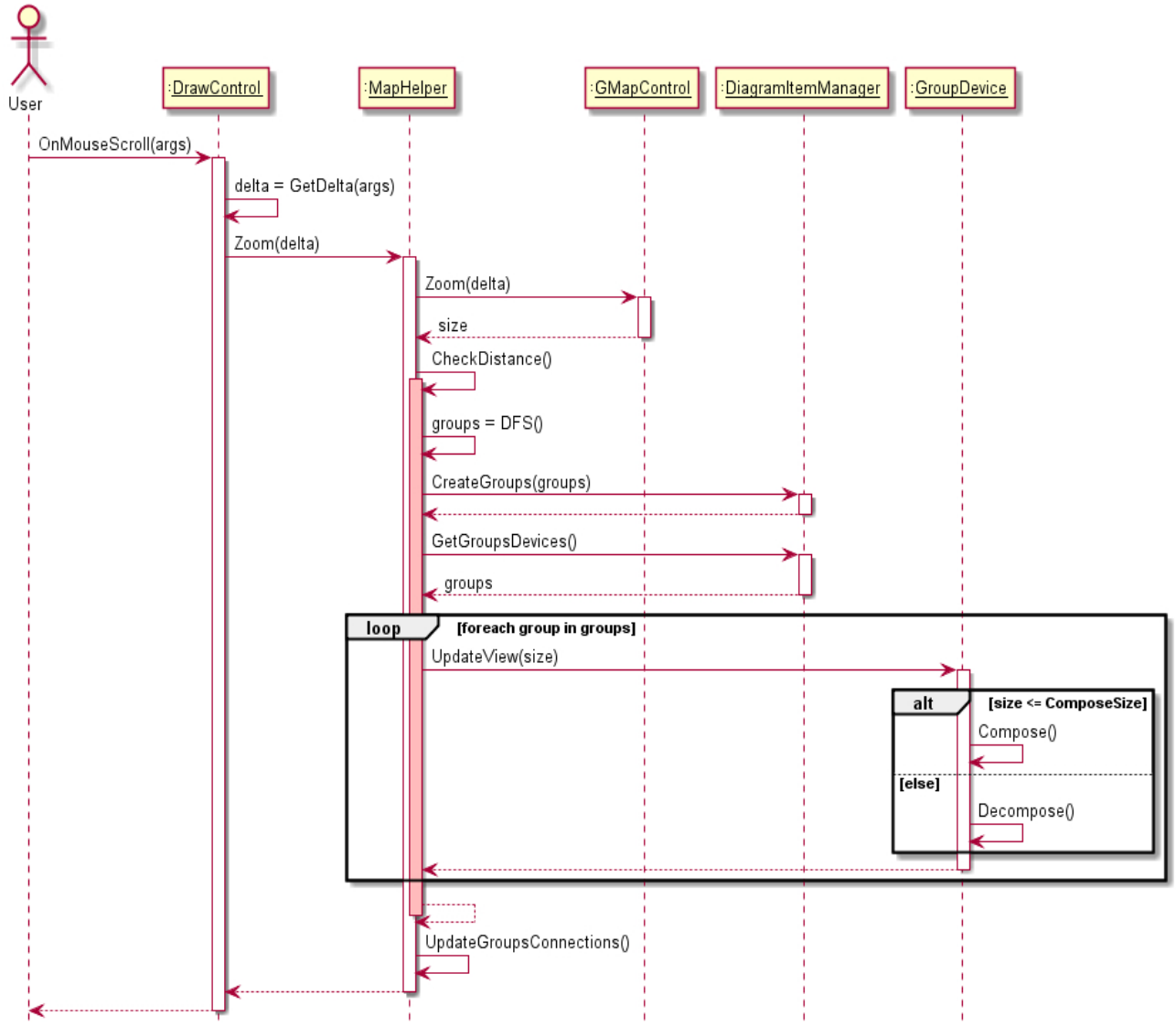

Рис. 5. Диаграмма последовательности изменения масштаба схемы.

Из методов элемента управления GMapControl, которые использовались в системе, можно выделить FromLocalToLatLng (перевод координат из локальных в географические) и FromLatLngToLocal (перевод координат из географических в локальные), благодаря которым

стало возможным расположение элементов на карте. Также использовался метод Zoom, позвлоляющий менять масштаб карты.

4. Реализация приложения

Визуальный конструктор был разработан на платформе .NET с помощью языка программирования C# [4]. В качестве графической подсистемы использовался WPF (Windows Presentation Foundation) [7]. Также используется бесплатная библиотека с открытым исходным кодом GMap.NET [5], предназначенная для работы с географическими картами.

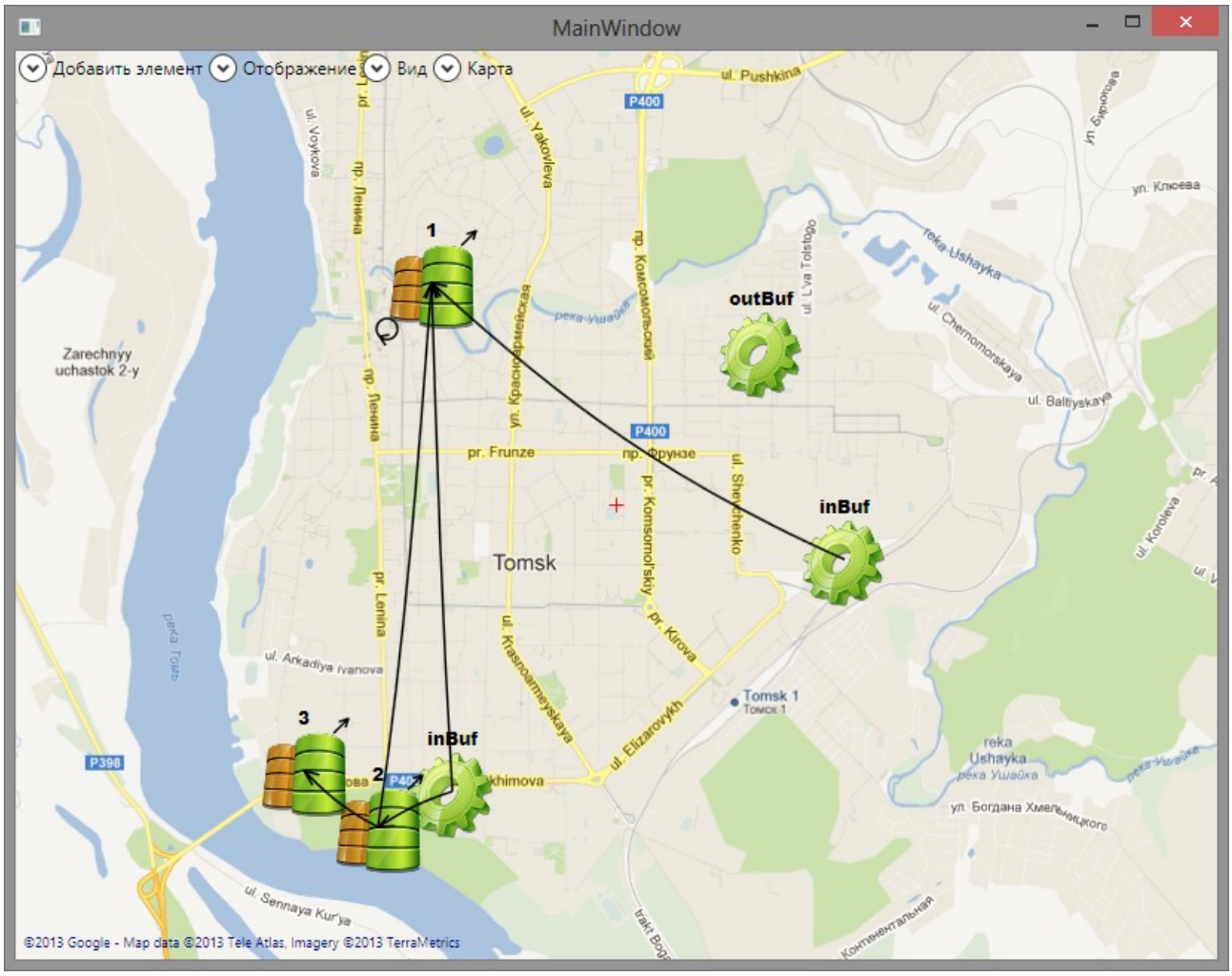

Рис. 6. Пример работы конструктора.

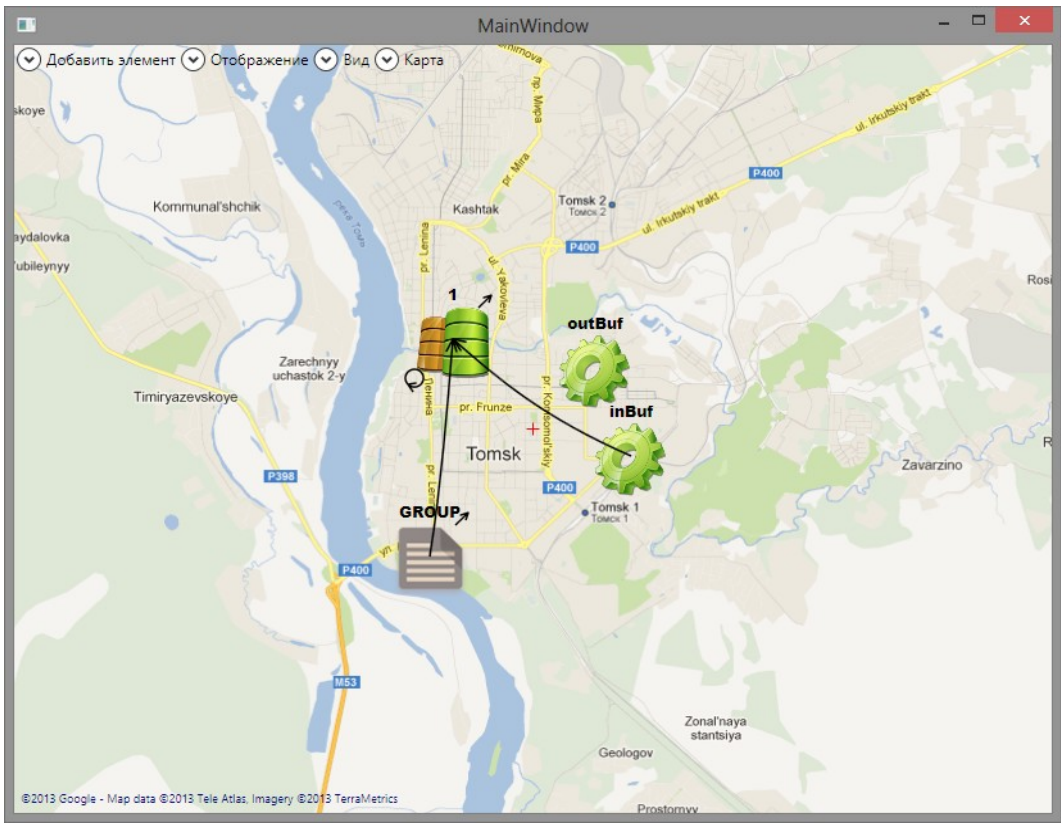

Рис. 7. Результат группировки, в случае отдаления карты на 1 шаг.

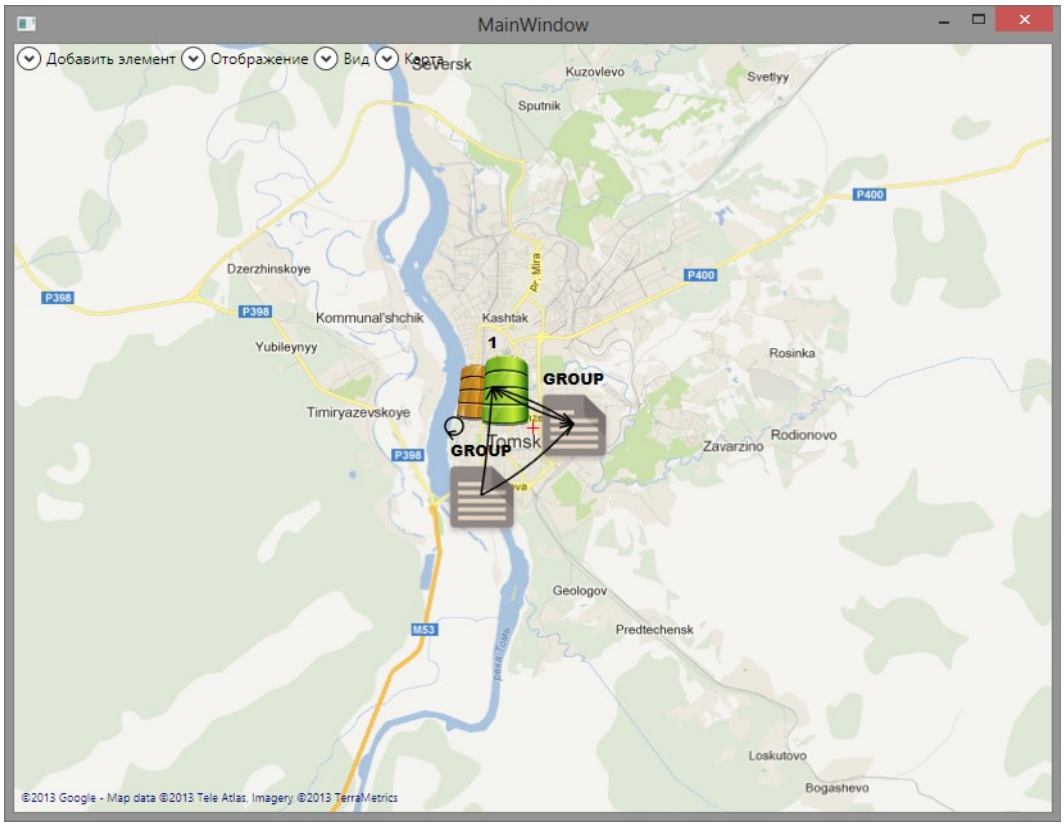

Рис. 8. Результат группировки, в случае отдаления карты на 2 шага.

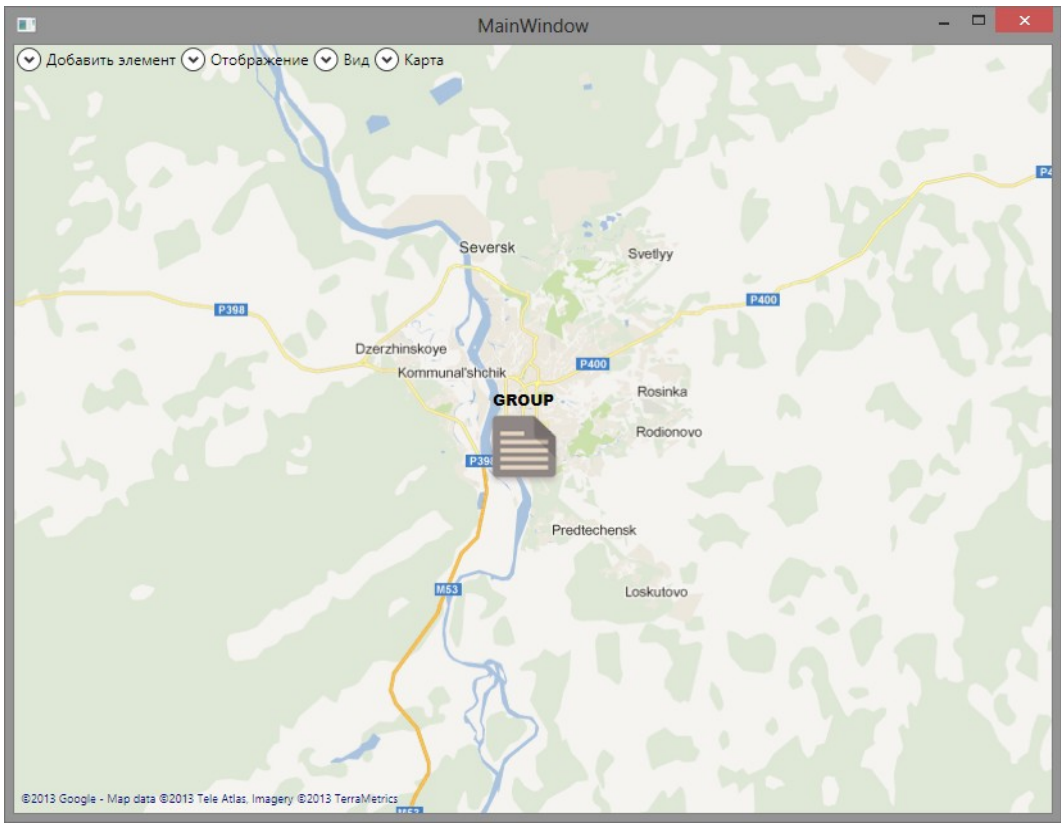

Рис. 9. Результат группировки, в случае отдаления карты на 3 шага.

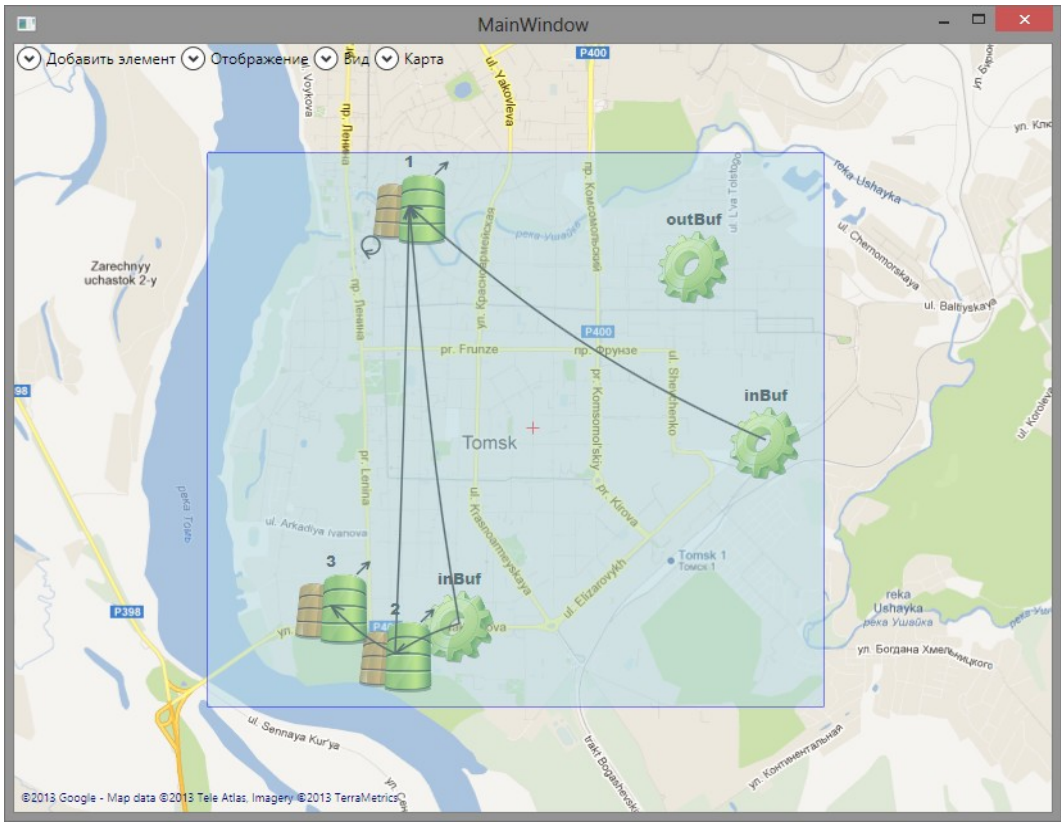

Рис. 10. Групповые операции над объектами.

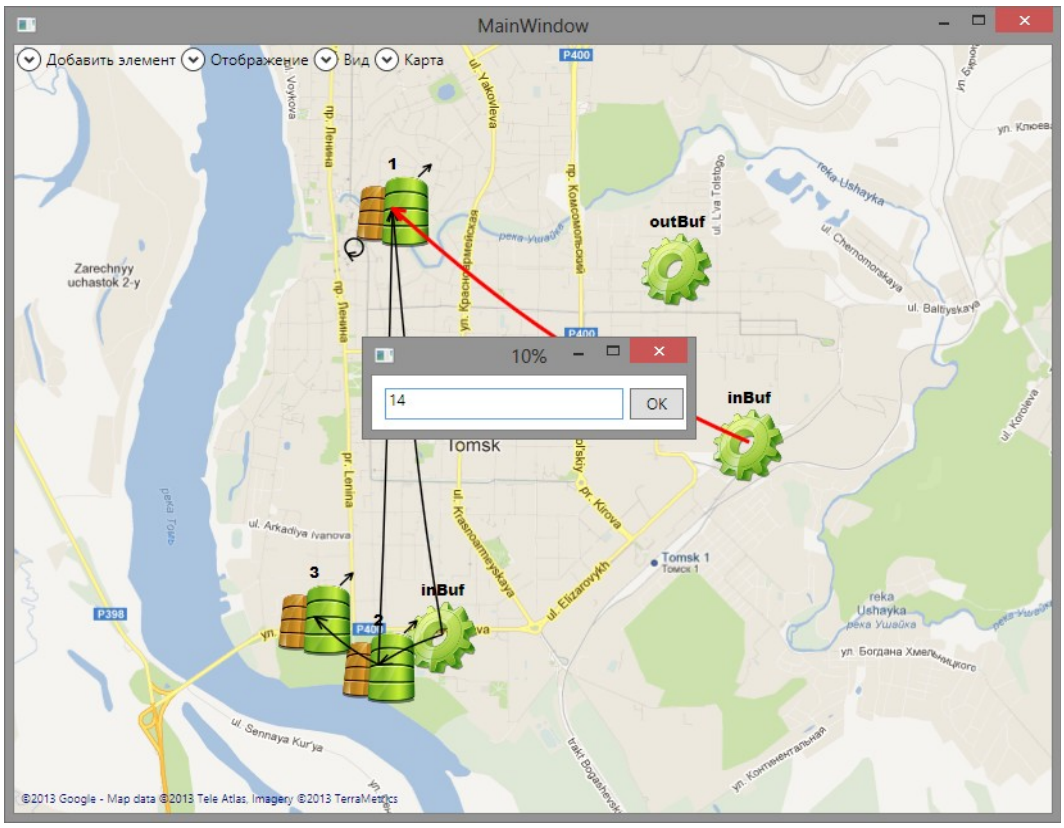

Рис. 11. Изменение характеристик системы (вероятность перехода).

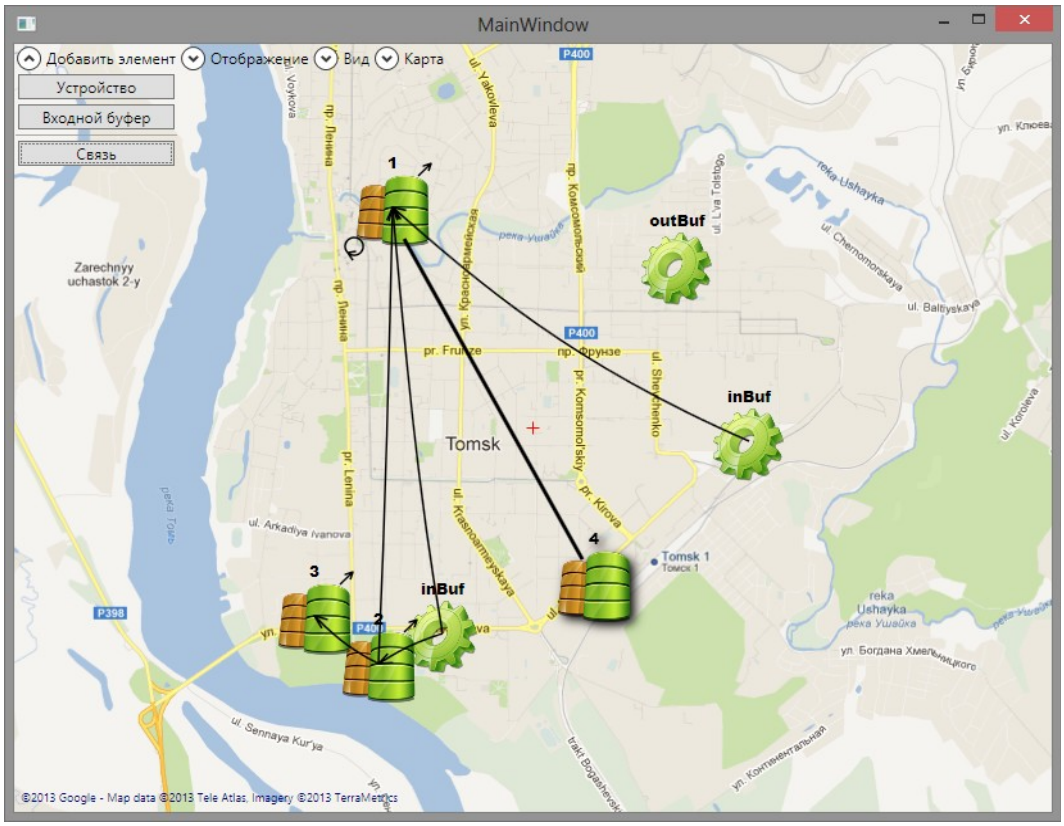

Рис. 12. Добавление нового блока обслуживающих приборов.

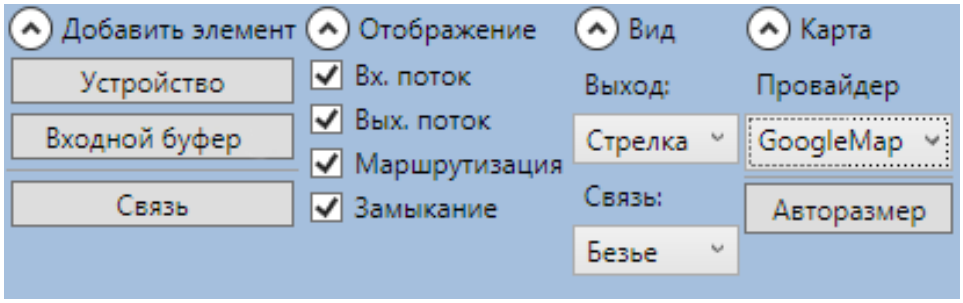

Рис. 13. Элементы управления.

Заключение

В рамках данной работы были решены все поставленные задачи.

Таким образом, цель — разработка визуального конструктора для системы имитационного моделирования процессов массового обслуживания — достигнута, приложение разработано и готово использованию.

Данная система применяется в комплекте программ ODIS в качестве визуального конструктора для системы имитационного моделирования процессов массового обслуживания.

В дальнейшем планируется вести работу по развитию конструктора в следующих направлениях:

- Добавление дополнительных возможностей по отображению объектов моделируемой системы массового обслуживания.
- Перевод на Web-платформу.
- Различные сервисные функции.

Литература

1. Гамма Э. Приемы объектно-ориентированного программирования. Паттерны проектирования: пер. с англ./ Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – СПБ : Питер, 2010. – 368.
2. Моисеев А., Моисеева С., Синяков М. Базовая объектная модель слоя предметной области системы имитационного моделирования процессов массового обслуживания // Application Of Information And Communication Technology In Economy And Education (ICAICTEE-2011), Proceedings Of The International Conference (December 2-3, 2011, Sofia, Bulgaria). - Sofia: University Of National And World Economy, 2011. - pp. 230 - 236.
3. Моисеев А.Н., Синяков М.В. Разработка объектно-ориентированной модели системы имитационного моделирования процессов массового обслуживания // Вестник Томского государственного университета. Управление, вычислительная техника и информатика – 2010, № 1, С. 89–93
4. C Sharp [Electronic resource] / Wikipedia. - 2013. - Electronic data. - URL: http://ru.wikipedia.org/wiki/C_Sharp (retrieval date: 14.01.2013)
5. GMap.NET [Electronic resource] / Codeplex. - 2013. - Electronic data. - URL: http://greatmaps.codeplex.com/ (retrieval date: 18.03.2013)
6. Nathan A. Windows Presentation Foundation Unleashed (WPF) / A. Nathan. - Sams Publishing, 2006, - P. 142 — 348
7. Windows Presentation Foundation [Electronic resource] / Microsoft. - 2012. - Electronic data. - URL: http://msdn.microsoft.com/ru-ru/library/ms754130.aspx (retrieval date: 13.11.2012)