



004616302

На правах рукописи

Черненко Виталий Евгеньевич

**НИЗКОУРОВНЕВОЕ ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ  
ТРАНСПОРТНЫХ СИСТЕМ**

Специальность 05.13.18 – математическое моделирование, численные методы и  
комплексы программ

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени  
кандидата технических наук

Ульяновск – 2010

- 2 ДЕК 2010

Работа выполнена на кафедре информационных технологий в Государственном образовательном учреждении высшего профессионального образования Ульяновский государственный университет

Научный руководитель: доктор технических наук, профессор  
Кумунжиев Константин Васильевич

Официальные оппоненты: доктор технических наук, профессор  
Варнаков Валерий Валентинович

кандидат технических наук, доцент  
Капитанчук Василий Вячеславович

Ведущая организация: ГОУ ВПО Ульяновский государственный  
технический университет

Защита диссертации состоится «23» декабря 2010 г. в 13:00 на заседании диссертационного совета Д 212.278.02 при Ульяновском государственном университете по адресу: г Ульяновск, Набережная реки Свияги, 106, корп. 1, ауд. 703.

С диссертацией можно ознакомиться в библиотеке Ульяновского государственного университета, с авторефератом – на сайте ВУЗа <http://www.uni.ulsu.ru>.

Отзывы на автореферат просим присылать по адресу: 432000, г. Ульяновск, ул. Л.Толстого, д. 42, УлГУ, Управление научных исследований.

Автореферат разослан «20» ноября 2010 г.

Ученый секретарь  
диссертационного совета



Волков М.А.

# Общая характеристика работы

## Актуальность исследования

Компьютерное моделирование становится распространенным средством анализа сложных систем. Современный рынок внедрения и сопровождения технологических систем часто требует разработки систем поддержки принятия стратегических и оперативных решений на основе имитационных моделей. Так, низкоуровневое имитационное моделирование (ИМ) все чаще применяется при принятии решений о проектировании и реорганизации транспортных систем<sup>1</sup> (ТС).

Выделяют два подхода к разработке имитационных моделей транспортных систем: использование сред моделирования общего назначения и проблемно-ориентированных инструментов<sup>2</sup>.

Использование систем общего назначения предоставляет разработчикам больше возможностей, однако обладает рядом ограничений:

- моделирование (например, в RePast, Simio) предполагает более глубокую декомпозицию моделируемого объекта, сведение его элементов и подсистем к сущностям используемого в инструменте языка моделирования, что мешает построить модель без привлечения специалистов по ИМ;
- инструменты не содержат конструктивных элементов, необходимых для моделирования транспортных систем, из-за чего, например, в TransportLibrary AnyLogic 5 необходимо явно указывать траектории движения агентов.

Проблемно-ориентированные системы лишены этих недостатков, но их использование сопряжено с другими трудностями:

- прикладные разработчики ограничены стандартным набором композиционных элементов (например, в SidraIntersection), расширение которого трудоемко и предполагает обращение к разработчикам инструмента;
- невозможно исследовать ТС в составе моделей других организационно-технических систем;

---

<sup>1</sup> Yafeng Yin, Henry X. Liu, Jorge A. Laval, Xiao-Yun Lu, Meng Li, Joshua Pilachowski, Wei-Bin Zhang Development of an Integrated Microscopic Traffic Simulation and Signal Timing Optimization Tool / University of California, Berkeley, 2007

<sup>2</sup> ZOU Zhi-jun, YANG Dong-yuan An Object-oriented Development of Urban Traffic Simulation Laboratory System / Journal of system simulation, vol. 14, no. 7, 2002, pp. 844-848

- исследователи также ограничены в возможностях представления процессов и результатов моделирования, не могут отойти от навязываемого системой подхода.

Таким образом, представляется целесообразным совместить эти два подхода, создав расширяемую систему низкоуровневого агентного имитационного моделирования транспортных узлов, ориентированную с одной стороны на специалистов предметной области, а с другой – на профессионалов-имитационщиков, имеющих возможность расширять ее функциональность.

Основной проблемой при разработке требуемой системы на основе среды моделирования общего назначения является отсутствие проблемно-ориентированного языка моделирования. Именно этот язык и служит основой ориентированности на специалистов предметной области. Предметная ориентированность, с одной стороны, и расширяемость – с другой должны стать основными факторами, принимаемыми во внимание при декомпозиции предметной области «транспортные системы».

Объектом исследования в данной работе является система низкоуровневого имитационного моделирования ТС и её язык.

Предметом исследования является внутренняя структура транспортных систем, формальное описание языка и архитектура программного комплекса для низкоуровневого моделирования ТС.

### **Цель и задачи работы**

Целью работы является расширение сферы применения моделирования транспортных систем за счет создания проблемно-ориентированного средства на основе средства моделирования общего назначения.

Для достижения поставленной цели в работе решены следующие задачи:

- Разработана структура системы низкоуровневого имитационного моделирования ТС.
- Проведена декомпозиция предметной области «транспортные системы», выявлены её особенности с точки зрения структуры имитационных моделей.
- Создан язык моделирования транспортных систем.
- Создан графический редактор конструкций языка и программный интерфейс для выполнения агентных моделей транспортных систем.

### **Методы исследования**

В ходе исследования применялись методы системного анализа, имитационного моделирования и теории графов. При программной реализации

системы моделирования использовались методы структурного и объектно-ориентированного программирования.

### **Научная новизна**

В работе предложена новая структура расширяемой системы низкоуровневого имитационного моделирования ТС. Предложенная структура позволяет комбинировать преимущества проблемно-ориентированных инструментов и средств общего назначения. На основе предложенного способа декомпозиции разработан язык моделирования ТС. Также разработан новый программный комплекс для низкоуровневого моделирования транспортных систем.

### **Основные положения, выносимые на защиту**

- Структура системы низкоуровневого имитационного моделирования ТС, обеспечивающая расширяемость и ориентированность на специалистов в предметной области.
- Метод декомпозиции транспортных систем, ориентированный на имитационное моделирование и позволивший систематизировать множество объектов транспортной инфраструктуры.
- Язык моделирования и численные методы анализа структуры моделей транспортных систем.
- Программный комплекс, состоящий из графического редактора моделей, программного интерфейса для исполнения агентных моделей транспортных систем и вспомогательных модулей.

### **Достоверность результатов**

Достоверность результатов, полученных в работе, достигается корректностью применения методов системного анализа, однозначностью разработанных алгоритмов и подтверждается результатами компьютерного моделирования и тестирования разработанного программного комплекса. Результаты исследования обсуждались на российских и международных конференциях.

### **Теоретическая и практическая значимость**

Проведенная декомпозиция предметной области «транспортные системы», и созданное формальное описание языка позволит использовать данные геоинформационных систем для построения имитационных моделей.

Разработан программный комплекс низкоуровневого имитационного моделирования транспортных систем, позволяющий анализировать свойства

существующих и проектируемых транспортных узлов. Комплекс совмещает достоинства средств моделирования общего назначения и проблемно-ориентированных инструментов. Созданный комплекс может быть использован в муниципальных образованиях, проектных организациях и консалтинговых компаниях, занимающихся проектированием и реорганизацией схем дорожного движения.

### **Апробация работы**

Основные результаты исследования были апробированы на конференциях:

- VII Международной конференции «Математическое моделирование физических, экономических, технических систем и процессов», Ульяновск, УлГУ, 2009;
- Interactive Systems and Technologies: the Problems of Human-Computer Interaction, Ulyanovsk, ULSTU, 2009;
- Четвертая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности (ИММОД-2009), Санкт-Петербург, 2009;
- Всероссийская конференция «Проведение научных исследований в области обработки, хранения, передачи и защиты информации» (ОИ-2009), Ульяновск, УлГТУ, 2009;
- Winter Simulation Conference (Ph. D. Colloquium), Austin, TX, USA, 2009;
- Информатика, моделирование, автоматизация проектирования. (ИМАП-2010), Ульяновск, 2010.

### **Личный вклад автора**

Постановка задачи выполнена автором самостоятельно при методической поддержке научного руководителя. Создание программного комплекса, разработка языка и структуры системы, а также тестирование выполнены автором самостоятельно.

### **Публикации**

Материалы диссертации опубликованы в 11 работах, из них 2 – в изданиях, рекомендуемых ВАК. Список публикаций приведен в конце автореферата.

### **Структура и объем диссертации**

Диссертация состоит из введения, четырех глав, заключения, списка используемой литературы из 87 наименований и приложений. Основной текст диссертации изложен на 107 страницах и содержит 35 рисунков и 6 таблиц.

## Содержание работы

**Введение** содержит общее описание работы, постановку цели и задач исследования, обоснование его актуальности. Также приведены основные результаты работы и варианты их практического применения.

В **первой главе** приводится обзор существующих языков моделирования, как общего назначения, так и проблемно-ориентированных. Отмечаются основные недостатки существующих систем моделирования с точки зрения удобства и трудоемкости разработки моделей.

Под системой моделирования в работе понимается совокупность языка разработки модели, программной системы исполнения модели и сопутствующей инфраструктуры, а под языком моделирования – совокупность элементов, описывающих базовые сущности моделей, и правил композиций этих элементов. Язык системы моделирования во многом схож с языками программирования, так как требует однозначности всех конструкций и строго следования синтаксическим правилам.

В современных средах моделирования наблюдаются две противоположные тенденции развития языков:

- стремление к наиболее обобщенному заданию математических и логических формализмов и, соответственно, к унификации процесса разработки моделей;
- возрастающая ориентация на конкретные предметные области, использование элементов языка, отражающих названия, форму и структуру конкретных объектов предметной области.

Современные проблемно-ориентированные системы моделирования имеют развитые подсистемы ввода моделей, позволяющие определять структуру модели в графическом режиме, приближая язык к предметной области, на которую они ориентируются.

В настоящее время существуют несколько средств низкоуровневого моделирования транспортных систем:

- VISSIM<sup>3</sup> – разработка немецкой фирмы PTV Systems. Позволяет строить низкоуровневые модели транспортных систем, состоящих из участков дороги, перекрестков любой формы, многоуровневых развязок. Основной особенностью является отображение трехмерной анимации в ходе моделирования.

---

<sup>3</sup> <http://www.vissim.de/>

- Transmodeler<sup>4</sup> – разработка фирмы Caliper (США). Позволяет интегрировать модели транспортных систем с данными геоинформационных систем и предназначена в основном для моделирования движения по шоссе и сложным развязкам. Ориентирована на особенности дорожного движения в США.
- SIDRA Intersection<sup>5</sup> – разработка фирмы SidraSolutions (Австралия). Позволяет моделировать поведение участников дорожного движения на перекрестках и прилегающих к ним территориях, обладает мощными средствами сбора и анализа статистики состояния модели.
- Transport Library AnyLogic 5 – разработка отечественной компании Экс Джей Текнолоджис. Главной особенностью является возможность сочетать модели транспортной системы с другими имитационными моделями. Обладает богатыми возможностями настройки анимации.

Моделирование в VISSIM и Transmodeler предоставляет широкий набор возможностей, но часто является неоправданно сложным и недоступным специалистам в предметной области, так как предполагает глубокое знание технологий имитационного моделирования. В TransportLibrary AnyLogic 5 необходимо детально указывать возможные траектории движения агентов, что делает невозможным, например, адекватно моделировать перестроения агентов при движении по многополосному участку дороги. Некоторые (SIDRA Intersection и другие) средства позволяют моделировать только отдельные типы объектов транспортной инфраструктуры.

По результатам обзора делается вывод о целесообразности создания новой системы, обладающей следующими свойствами:

- простота создания моделей для специалистов предметной области;
- расширяемость моделей, предоставление возможности инженерам-программистам дополнять набор средств разработки моделей;
- наглядность моделирования, максимальное визуальное сходство модели и моделируемой системы.

Во **второй главе** предлагается структура системы и описываются сценарии ее использования.

В качестве основы системы была выбрана среда моделирования AnyLogic 6. Выбор обусловлен наличием в ней следующих возможностей:

- богатый набор инструментов для визуализации моделей;

<sup>4</sup> <http://www.caliper.com/transmodeler/>

<sup>5</sup> <http://www.sidrasolutions.com/>

- возможность проводить эксперименты типа «что-если?» и оптимизационные эксперименты по подбору параметров;
- обеспечение расширяемости за счет добавления новых модулей в подключаемых библиотеках;
- широкие возможности по обмену данными с внешними приложениями, такими как MS Excel и базами данных.

Структура системы моделирования ТС показана на рисунке 1.

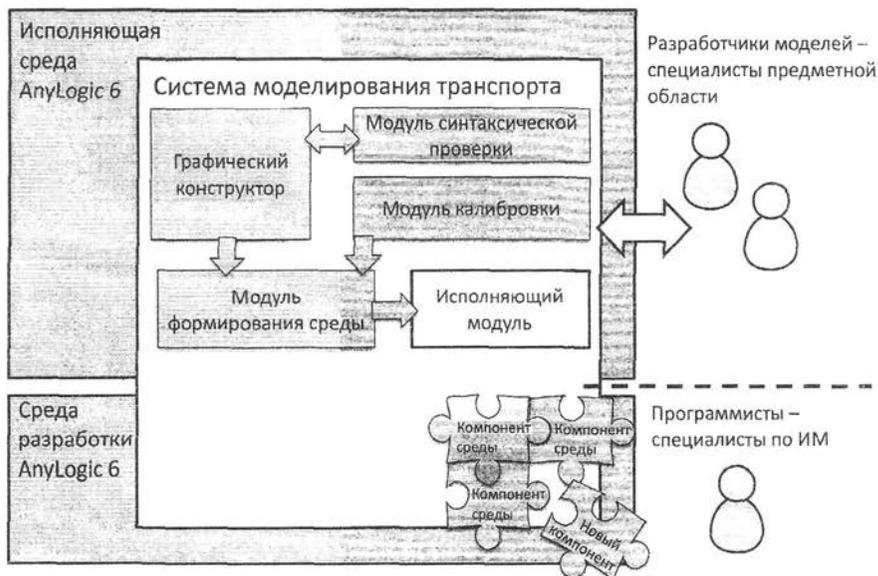


Рисунок 1. Структура системы моделирования ТС

Предложено реализовать основную часть системы как отдельное Java-приложение, выполняемое на основе имитационной среды AnyLogic. Имитационная среда AnyLogic, согласно предложенной структуре, является ядром всей системы: она предоставляет базовую функциональность по исполнению имитационной модели.

Общая схема процесса взаимодействия прикладного разработчика моделей с системой показана на рисунке 2. Также на рисунке показана последовательность исполнения разработанных модулей системы.

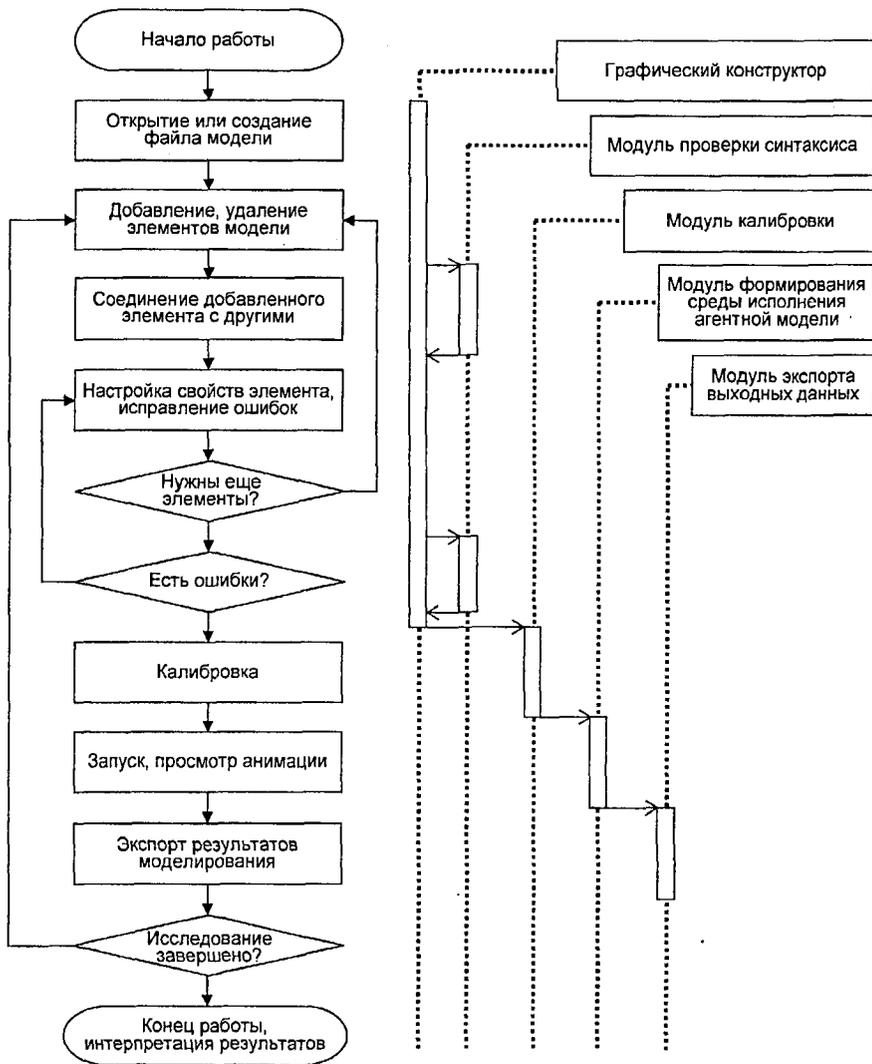


Рисунок 2. Схема процесса взаимодействия пользователя с системой

Прикладные разработчики создают модели с помощью графического конструктора и имеют возможность сохранять их в файлах формата XML.

Графический конструктор содержит модуль проверки синтаксиса разрабатываемых моделей, который указывает на синтаксические ошибки,

возникающие в ходе разработки модели. Запуск моделирования возможен только при условии отсутствия ошибок.

С помощью модуля калибровки исследователи настраивают параметры участников дорожного движения в соответствии с данными, полученными на участках, аналогичных моделируемому.

Модуль формирования среды обитания агентов транслирует заданную пользователем конфигурацию транспортной системы во внутреннее представление, совместимое с исполняющим ядром AnyLogic и предназначенное для исполнения агентной модели. Исполняющий модуль обеспечивает возможность запуска экспериментов в любом желаемом масштабе времени. Главной частью исполняющего модуля является анимация модели, демонстрирующая изменение состояния модели во времени.

По окончании моделирования отчет об эксперименте может быть выведен во внешнее приложение для последующего анализа. Интеграцию экспериментальных данных с внешними хранилищами обеспечивает модуль экспорта выходных данных.

В режиме исполнения модели отображается анимация, представляющая собой двухмерный план моделируемой системы с движущимися по ней транспортными средствами (рисунок 3).

При создании моделей реально существующих транспортных систем неизбежны ситуации, когда набор существующих компонентов системы не позволяет отразить структуру моделируемого транспортного узла. Возникает необходимость добавления новых компонентов для наиболее точного задания структуры моделируемой системы. В таких случаях исследователи транспортной системы вынуждены обращаться к специалистам в области имитационного моделирования, обладающим навыками программирования в среде AnyLogic 6.

Такое взаимодействие, схематично изображенное на рисунке 4, предусмотрено на уровне структуры созданной среды.

Разработкой нового компонента системы для моделирования нестандартного транспортного узла занимается организация, специализирующаяся на имитационном моделировании. В результате система дополняется новым компонентом, который может быть использован не только в конкретном проекте, но и для моделирования других подобных транспортных узлов.

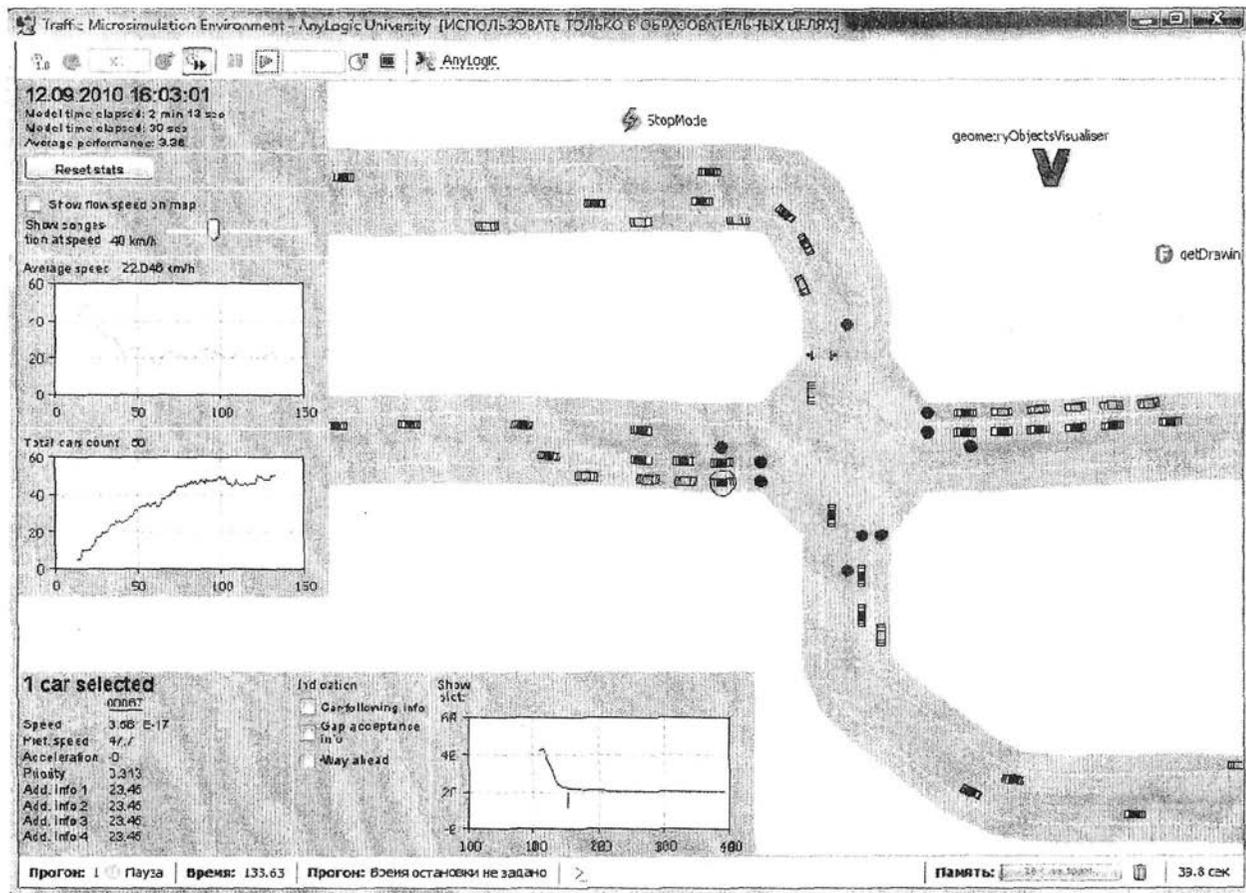


Рисунок 3. Внешний вид анимации эксперимента



Рисунок 4. Схема взаимодействия исследователей и специалистов по ИМ

При разработке новых компонентов специалисты по имитационному моделированию могут пользоваться всем набором средств, предоставляемых системой AnyLogic 6 (диаграммы состояний, события, низкоуровневая реализация логики на языке Java).

Разработанная система может быть использована для поддержки принятия решений и охватывает все этапы проведения исследований с помощью имитационного моделирования.

В третьей главе описывается разработанный язык моделирования транспортных систем. Основные положения, принятые во внимание при декомпозиции предметной области «транспортные системы», ориентированной на низкоуровневое имитационное моделирование, следующие:

- Учет мелкого масштаба моделирования, необходимости рассматривать поведение отдельных участников движения и геометрию участков дороги.
- Направленность на отделение активных, действующих элементов системы от пассивных, влияющих на моделирование посредством наложения ограничений на активные элементы.

На основании этих положений предложены принципы структурной декомпозиции транспортных систем для мелкомасштабного агентного имитационного моделирования:

- Транспортная система состоит из двух основных элементов – множества отдельных взаимодействующих между собой участников движения и множества объектов транспортной инфраструктуры, воздействующих на участников движения.
- Множество объектов транспортной инфраструктуры разбито на подмножества, каждому из которых соответствует определенная модель (тип алгоритма) поведения участников движения.

Декомпозиция транспортной системы предполагает выделение участков, обладающих сходными алгоритмами поведения участников движения. Такая декомпозиция позволяет выделить несколько типов структурных элементов транспортных систем (таблица 1).

Таблица 1. Типы элементов языка моделирования транспортных систем

Тип	Типичные представители	Основные алгоритмы поведения участников движения
Участок дороги	прямолинейный участок, участок с изгибами, мост, тоннель, эстакада	выбор предпочтительной полосы движения, оценка возможности перестроения, поддержание безопасной дистанции до впереди идущего агента
Пересечение	перекресток, примыкание дороги, круговой перекресток	выбор траектории маневра, объезд возможных препятствий, приоритетный разезд
Сложный объект	заправочная станция, парковка, банк или магазин типа «drive in»	остановка на заданное время на определенном месте, маневрирование на низкой скорости, ожидание в очереди, приоритетный разезд

Необходимым типом элементов языка моделирования транспортных систем является тип «Вспомогательный элемент». Элементы данного типа не соответствуют никаким реально существующим объектам, но необходимы для корректного построения имитационной модели.

Например, при моделировании ограниченного участка дороги необходимо указать интенсивность потоков на границах моделируемого участка, что обусловило необходимость введения элемента «Граница моделируемого участка», относящегося к дополнительному типу «Вспомогательный элемент». Общая структура реализованного подмножества предложенного языка показана на рисунке 5.

Специфика разработанного языка предполагает наличие удобного редактора, позволяющего создавать конструкции в режиме визуального редактирования, поэтому каждый элемент графического языка имеет визуальное представление, являющееся набором графических примитивов. Это представление используется в графическом конструкторе, предназначенном для создания моделей транспортных систем. Визуальные представления некоторых элементов проиллюстрированы на рисунке 6: перекресток (А), многополосный участок дороги (Б) и граница моделируемого участка (В).

Каждый тип элементов характеризуется собственным набором параметров. Например, для перекрестка характерны количество полос движения, наличие и ширина разделительной полосы каждой из примыкающих дорог; для

участка дороги – длина, конфигурация изгиба, количество полос движения в каждом направлении, наличие и ширина разделительной полосы; для границы моделируемого участка – количество полос движения в каждом направлении и наименование.

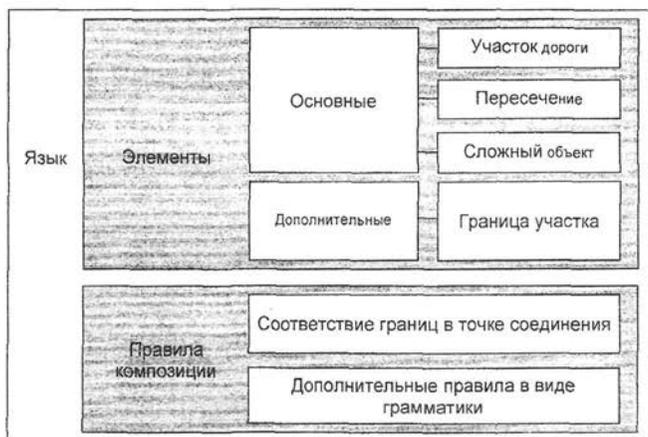


Рисунок 5. Структура реализованного языка моделирования

Помимо собственных параметров каждый элемент характеризуется параметрами, задающими его положение в двухмерном пространстве. Для однозначности задания такого положения выбран следующий метод: для каждого элемента определена начальная точка; положение элемента в пространстве и его поворот заданы относительно этой точки. Каждая манипуляция в графическом конструкторе соответствует изменению некоторого параметра одного или нескольких элементов. Например, операция перетаскивания приводит к изменению положения фигуры в двумерном пространстве.

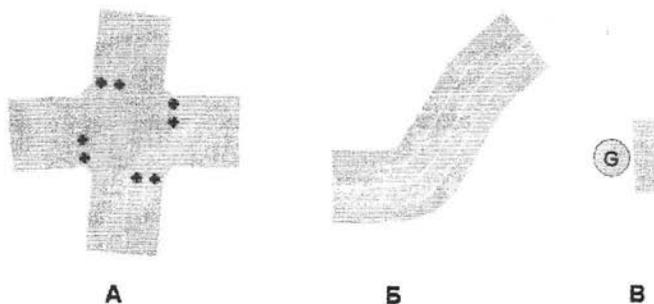


Рисунок 6. Визуальное представление элементов языка

Модель состоит из множества конструкционных элементов

$$M = \{E_1, \dots, E_N\}.$$

Конструкционные элементы могут принадлежать к любому из трех типов, однако, они предназначены для функционирования в единой структуре модели, поэтому целесообразно хранить как можно больше информации в базовом классе ConstructionElement. Такая необходимость требует обобщения структуры всех элементов.

Каждому элементу модели  $E_i$  соответствует граф

$$E_i = \langle V_i = \{V_i^1, \dots, V_i^M\}, R_i = \{R_i^1, \dots, R_i^K\} \subset P_i^2 \rangle,$$

также каждому элементу модели сопоставлено  $V_i^* \subset V_i$ , то есть подмножество внешних вершин графа элемента. Внешние вершины предназначены для соединения элементов. Каждая внешняя вершина характеризуется дескриптором вида  $D(V_i^k) = \begin{cases} \emptyset, & \text{если } V_i^k \notin V_i^* \\ \alpha \in N^3, & \text{если } V_i^k \in V_i^* \end{cases}$ . Тогда в случае если все элементы корректно соединены, то есть  $\forall i, k: D(V_i^k) \neq \emptyset \exists j, t: i \neq j, V_i^k = V_j^t$  граф транспортной системы получается путем объединения графов всех элементов, а именно:

$$G = \langle \bigcup_{i=1}^N V_i, \bigcup_{i=1}^N R_i \rangle$$

Обобщение элементов в виде подграфов транспортной системы позволяет:

- хранить минимум информации о модели, полностью отделив существенные данные от визуального представления;
- унифицировать разработку новых конструкционных элементов за счет проведенного обобщения их структуры.

Способом композиции элементов в разработанном языке является их соединение. Каждый элемент имеет специальные точки соединения. Точка соединения характеризуется координатой  $T$  и углом направления  $\alpha$ . Формулы (1) определяют условия соединения двух элементов (рисунок 7А).

$$\begin{cases} T_1 = T_2 \\ \alpha_1 = -\alpha_2 \\ n_1 = m_2 \\ m_1 = n_2 \end{cases} \quad (1)$$

Где  $T_1, T_2$  – точки соединения элементов  $E_1$  и  $E_2$ ,  $\alpha_1, \alpha_2$  – углы направления соединения точек  $T_1$  и  $T_2$ ,  $n_1, n_2$  – количество входящих полос движения

элементов  $E_1$  и  $E_2$ ,  $m_1$ ,  $m_2$  – количество исходящих полос движения элементов  $E_1$  и  $E_2$ .

Таким образом, соединение элементов налагает на них определенные ограничения (в частности, ограничения на расположение, поворот, количество полос). Это позволяет говорить о синтаксических правилах композиции элементов в разработанном языке.

Настройка параметров двух объектов, необходимых для их соединения, может стать трудоемкой для разработчика модели. Для снижения трудоемкости использования редактора был предложен упрощенный механизм соединения элементов: в графический редактор была добавлена новая сущность – соединительный порт. Он характеризуется ссылкой на объект, к которому порт принадлежит, и соответствующей ему точкой соединения. При соединении один из соединяемых элементов является независимым, а другой – зависимым. В процессе соединения свойства зависимого элемента автоматически изменяются с учетом свойств независимого таким образом, чтобы обеспечивались условия соединения объектов. Чтобы соединить два элемента, пользователю необходимо:

- определить зависимый и независимый элементы;
- поднести путем перетаскивания зависимый элемент таким образом, чтобы его свободный порт оказался на расстоянии не более 20 пикселей от свободного порта независимого элемента;
- закончить перетаскивание, при этом свойства зависимого элемента автоматически изменятся, и два объекта станут соединенными.

Рисунок 7 (Б, В) иллюстрирует положение двух объектов до и после операции соединения.

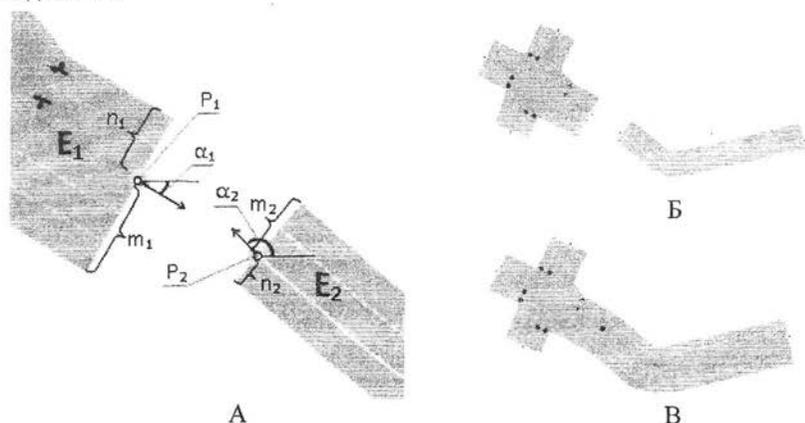


Рисунок 7. Соединение двух конструктивных элементов

Таким образом, в графическом конструкторе реализуются правила соединения элементов языка. Однако даже при соблюдении этих правил конструкция языка может быть бессмысленной: появляется необходимость выработки дополнительных правил композиции элементов на более высоком уровне, которые обеспечили бы логическую непротиворечивость построенной модели.

Эти дополнительные правила удобно определять с помощью графовых грамматик. Графовая грамматика может быть представлена множеством продукций вида  $P_i: G \rightarrow G^*$ , где  $i$  – номер продукции, а  $G$  и  $G^*$  – графы в левой и правой ей частях. Для конструкций разработанного языка были созданы 8 продукций, показанных в таблице 2. Причем,  $S, A, B, R$  – нетерминальные символы,  $E, R^*, C$  – терминальные. Продукция номер 8 содержит специальный элемент, необходимый вследствие того, что анализируемые графы могут содержать циклы. Элемент в правой части этой продукции означает полностью разобранный подграф.

Таблица 2. Схема конечного автомата анализа конструкций языка

№	Продукция	№	Продукция
1		5	
2		6	
3		7	
4		8	

Работа конечного автомата при анализе конструкции состоит в последовательном применении продукций 1-8 к анализируемому графу. В результате применения продукции к графу он разбивается на несколько подграфов, к которым процедура анализа применяется рекурсивно. В память конечного автомата записываются все посещенные при разборе узлы графа. Анализ считается успешным, если на всех его итерациях удалось разбить

анализируемый подграф согласно одной из схем 1–8. Работа конечного автомата описывается алгоритмом (листинг 1).

```
системаПродукций = {продукцияi = {исходныйГрафi -> множествоПодграфовi}, i=1..8};
помеченныеУзлыГрафа = ∅;
терминальныеСимволы = { E, R*, C };
символПомеченногоЭлемента = X;
function синтаксическийАнализ (анализируемыйПодграф) begin
  for each терминальныйСимвол in терминальныеСимволы do begin
    if (соответствуетСхеме (анализируемыйПодграф, терминальныйСимвол)) then begin
      помеченныеУзлыГрафа = помеченныеУзлыГрафа ∪ анализируемыйПодграф.узлы(0);
      return true;
    end
  end
  if ( соответствуетСхеме (анализируемыйПодграф, символПомеченногоЭлемента) and
      анализируемыйПодграф.узлы(0) ⊆ помеченныеУзлыГрафа ) then begin
    return true;
  end
  for each продукция in системаПродукций do begin
    if (соответствуетСхеме (анализируемыйПодграф, продукция)) then begin
      множествоПодграфов = разделитьНаПодграфы (анализируемыйПодграф, продукция);
      for each подграф in множествоПодграфов do begin
        if (not(синтаксическийАнализ (подграф))) then return false;
      end
      return true;
    end
  end
  return false;
end
```

*Листинг 1. Алгоритм работы конечного автомата синтаксического анализа конструкций языка*

В четвертой главе приводится описание программной реализации предложенного языка и связанных с ним элементов программного комплекса. Приводится описание реализованных алгоритмов, описываются отдельные структурные решения и разработанная библиотека пользовательского ввода.

Перед началом моделирования запускается рекурсивный алгоритм, производящий обход графа копструкций языка в глубину. Результатом работы алгоритма является конвертация графа (рисунок 8), состоящего из соединенных элементов языка в другой граф, в узлах которого находятся точки изменения алгоритмов поведения агентов, а ребра содержат всю необходимую информацию, используемую алгоритмом поведения агента на данном участке.

Таким образом, разработанный алгоритм формирует среду обитания агентов, являющуюся необходимой для функционирования агентов-участников движения -- активных объектов AnyLogic определенного типа.

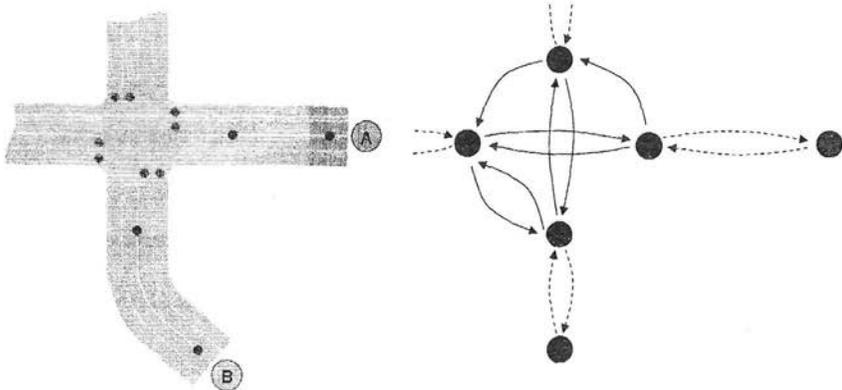


Рисунок 8. Построение графа фрагмента транспортной системы

Также описаны архитектурные решения по разработке графического конструктора, позволившие преодолеть следующие ограничения, налагаемые средой AnyLogic:

- отсутствие возможности наследования графических элементов активными объектами AnyLogic;
- выполнение отрисовки элементов презентации AnyLogic в недоступном для разработчика потоке;
- отсутствие возможности обрабатывать сложный пользовательский ввод.

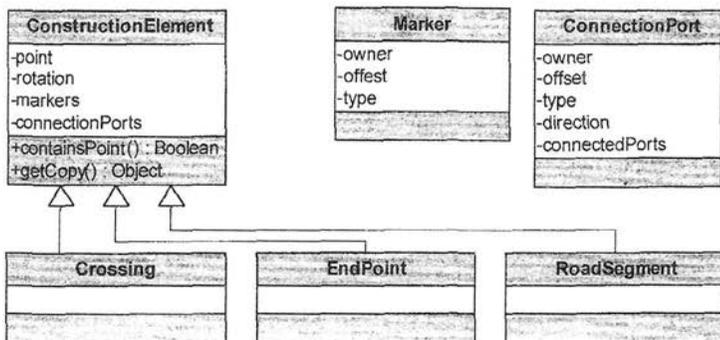


Рисунок 9. Диаграмма классов графического конструктора

Конструктор состоит из активных объектов AnyLogic и дополнительных дескрипторов, реализованных в виде Java классов. Диаграмма классов созданного графического конструктора в нотации UML приведена на рисунке 9.

Программная реализация графического конструктора обладает следующими особенностями:

- для упрощения настройки свойств конструкционных элементов была введена дополнительная сущность – служебный маркер – участок экрана, специальным образом обрабатывающий манипуляции мыши;
- существенные данные, описывающие состояния конструкционных элементов, были отделены от презентационных данных, все действия по отрисовке элементов языка были вынесены в функцию, выполняемую в отдельном потоке;
- диспетчеризация событий пользовательского ввода была перенесена на уровень обобщенного конструкционного элемента (класс `ConstructionElement`), что упростило расширение системы новыми элементами языка;
- добавление новых конструкционных элементов облегчается с помощью вынесения многих функций в базовый класс `ConstructionElement`.

Для обработки событий пользовательского ввода была разработана библиотека `AnyLogic`. Необходимость ее разработки обусловлена отсутствием поддержки многих функций пользовательского ввода, высокой вероятностью необходимости одних и тех же функций в разных проектах и, следовательно, возможностью повторного использования.

Библиотека, получившая название `InputExtensionLibrary`, содержит два класса активных объектов:

- класс «`KeyboardManager`», который при добавлении на презентацию некоторого активного объекта `AnyLogic` позволяет обрабатывать события клавиатуры для презентации этого объекта;
- класс «`MouseManager`», снабжающий активные объекты `AnyLogic` возможностью обработки событий мыши.

Библиотека «`InputExtensionLibrary`» реализована в виде подключаемой библиотеки `AnyLogic` в соответствии с требованиями, выдвигаемыми производителями. Подключение библиотеки и её использование производится с помощью стандартных операций среды `AnyLogic`.

В заключении сформулированы основные теоретические и практические результаты, дана оценка их применимости, а также приведены возможные направления дальнейших исследований. Основные результаты работы:

- Создана структура системы низкоуровневого имитационного моделирования ТС, ориентированной на специалистов в предметной области, обеспечивающая расширяемость за счет возможности добавления новых модулей

- Проведена декомпозиция предметной области «транспортные системы», ориентированная на имитационное моделирование и позволившая систематизировать множество объектов транспортной инфраструктуры.
- Разработаны язык описания и методы анализа транспортных систем, обеспечивающие возможность задания общей структуры среды взаимодействия агентов в низкоуровневых имитационных моделях ТС.
- Реализован программный комплекс, состоящий из графического редактора моделей, программного интерфейса для выполнения агентных моделей транспортных систем и вспомогательных модулей.

## **Список публикаций по теме диссертации**

### **Публикации в изданиях из списка ВАК**

1. Малыханов А.А., Черненко В.Е. Среда низкоуровневого имитационного моделирования транспортных систем // Автоматизация в промышленности №1-2010 – М., 2010 – С. 34–37
2. Малыханов А.А., Черненко В.Е., Былина П.В. Оценка эффективности траекторий патрулирования акватории на основе имитационной модели // Автоматизация процессов управления, №2-2010 – Ульяновск – С. 31–33

### **Другие публикации**

3. Черненко В.Е., Малыханов А.А. Библиотека геометрических объектов для AnyLogic 6 // Ученые записки УлГУ, серия Математика и информационные технологии – № 1(2). – Ульяновск: УлГУ – 2009 – С. 179–181
4. Кумунжиев К.В., Малыханов А.А., Черненко В.Е. Разработка системы имитационного моделирования транспортных узлов // Ученые записки УлГУ, серия Математика и информационные технологии – вып. 1(2). – Ульяновск: УлГУ, 2009 – С. 108–110
5. Малыханов А.А., Черненко В.Е. Классификация агентных имитационных моделей // Труды VII Международной конференции «Математическое моделирование физических, экономических, технических систем и процессов» – Ульяновск: УлГУ – 2009 – С. 179–181
6. Черненко В.Е., Малыханов А.А. Алгоритм построения траектории движения агента в модели транспортной системы // Труды VII Международной конференции «Математическое моделирование физических, экономических, технических систем и процессов» – Ульяновск: УлГУ, 2009 – С. 279–280

7. A.A. Malykhanov, K.V. Kumunjiev, V.E. Chernenko. Modeling Driver Behavior for Microscopic Traffic Simulation // *Interactive Systems and Technologies: the Problems of Human-Computer Interaction. Volume III. – Collection of scientific papres. – Ulyanovsk: ULSTU – 2009 – P. 402–407*
8. Малыханов А.А., Кумунжиев К.В., Черненко В.Е. Среда имитационного моделирования транспортных систем // Четвертая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности (ИММОД-2009). – СПб.: ЦНИИТС – 2009 – С. 284–287
9. Черненко В.Е. Графический конструктор моделей транспортных систем // Труды всероссийской конференции «Проведение научных исследований в области обработки, хранения, передачи и защиты информации» (ОИ-2009) – Ульяновск: УлГТУ – 2009 – т. 1 – С. 180–185
10. A. A. Malykhanov, V. E. Chernenko. Extensible Framework for Microscopic Traffic Simulation. Proceedings of the 2009 Winter Simulation Conference (Ph. D. Colloquium) – Austin, TX, USA – ISBN: 978-1-4244-5772-4
11. Черненко В.Е., Малыханов А.А. Анализ зависимости пропускной способности регулируемого участка от длительности светофорных фаз. // Информатика, моделирование, автоматизация проектирования. Сборник научных трудов школы-семинара ИМАП-2010 – Ульяновск – 2010 – С. 524–527

20

Подписано в печать 18.11.2010.  
Формат 60x84/16. Усл. печ. л. 1,0.  
Бумага книжно-журнальная. Гарнитура Times New Roman.  
Тираж 100 экз. Заказ № 129/650

Отпечатано с оригинал-макета в Издательском центре  
Ульяновского государственного университета  
432000, г. Ульяновск, ул. Л. Толстого, 42