МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

**Дробнов С.Е., Кошкин Д.Е.**

**МОДЕЛИРОВАНИЕ СИСТЕМ**

**практикум**

**ДЛЯ СТУДЕНТОВ, ОБУЧАЮЩИХСЯ ПО НАПРАВЛЕНИЯМ ПОДГОТОВКИ 38.03.04 и 38.03.05**

Москва – 2016

УДК 004.942

ББК 052-05

Дробнов С.Е., Кошкин Д.Е. МОДЕЛИРОВАНИЕ СИСТЕМ

**[Электронный ресурс]** Методические указания по выполнению лабораторных работ по дисциплинам «Имитационное моделирование» и «Математические методы моделирования информационных систем» для студентов, обучающихся по направлениям подготовки 38.03.04 «Государственное и муниципальное управление» и 38.03.05 «Бизнес-информатика» / Дробнов С.Е., Кошкин Д.Е. - М., МИРЭА, 2016. - 1 электрон.опт. диск (CD-ROM).

Методические указания содержат справочную информацию по вопросам моделирования в среде AnyLogic, а также рекомендации и задания для выполнения лабораторных работ.

Редактор: Миронов А.А. Минимальный системные требования:

Поддерживаемые ОС: Windows 2000 и выше Память: ОЗУ 128МБ

Жесткий диск: 20 Мб

Устройства ввода: клавиатура, мышь

Дополнительные программные средства: программа AdobeReader

Подписано к использованию по решению редакционно- издательского совета Московского технологического университета: 15.05.2016

Обьем: 2,2мб Тираж: 10

© Дробнов С.Е., Кошкин Д.Е., 2016

© МИРЭА, 2016

# ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_TOC_250008)

1. [НАЗНАЧЕНИЕ ПАКЕТА ANYLOGIC 5](#_TOC_250007)
2. ОПИСАНИЕ ИНТЕРФЕЙСА ПРОГРАММЫ

ANYLOGIC 5

1. [ОСНОВНЫЕ СПОСОБЫ РАБОТЫ С МОДЕЛЬЮ 9](#_TOC_250006)
   1. [Работа со схемой модели 9](#_TOC_250005)
   2. [Работа с анимацией модели 11](#_TOC_250004)
   3. [Работа с кодом модели 15](#_TOC_250003)
2. [ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ 16](#_TOC_250002)
3. [ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ 28](#_TOC_250001)
4. [КОНТРОЛЬНЫЕ ВОПРОСЫ 30](#_TOC_250000)

# ВВЕДЕНИЕ

*Опасность не в том, что компьютер однажды начнет мыслить, как человек, а в том, что человек однажды начнет*

*мыслить, как компьютер. Сидни Дж. Харрис*

Моделирование является основным методом исследований во всех областях знаний и методом оценок характеристик сложных систем, используемым для оценки и принятия решений в самых разных сферах человеческой деятельности.

В настоящее время нельзя назвать область человеческой деятельности, в которой не использовались бы методы моделирования. Особенно это относится к сфере управления различными системами принятия решений на основе получаемой информации.

Имитационное моделирование (ситуационное моделирование) – частный случай математического моделирования, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности. Такую модель можно «проиграть» во времени как для одного испытания, так и заданного их множества. При этом результаты будут определяться случайным характером процессов. По этим данным можно получить достаточно устойчивую статистику.

Имитационное моделирование – это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, с которой проводятся эксперименты с целью получения информации об этой системе. Экспериментирование с моделью называют имитацией (имитация – это постижение сути явления, не прибегая к экспериментам на реальном объекте).

К имитационному моделированию прибегают, когда:

* дорого или невозможно экспериментировать на реальном объекте;
* невозможно построить аналитическую модель: в системе есть время, причинные связи, последствие, нелинейности, стохастические (случайные) переменные;
* необходимо сымитировать поведение системы во времени.

Цель имитационного моделирования состоит в воспроизведении поведения исследуемой системы на основе результатов анализа наиболее существенных взаимосвязей между еѐ элементами или другими словами — разработке симулятора исследуемой предметной области для проведения различных экспериментов.

AnyLogic – программное обеспечение для имитационного моделирования бизнес-процессов, разработанное российской компанией «Экс ДжейТекнолоджис» (англ. XJ Technologies). Инструмент обладает современным графическим интерфейсом и позволяет использовать язык Java для разработки моделей.

AnyLogic включает в себя графический язык моделирования, а также позволяет пользователю расширять созданные модели с помощью языка Java. Интеграция компилятора Java в AnyLogic предоставляет более широкие возможности при создании моделей, а также создание Java апплетов, которые могут быть открыты любым браузером. Эти апплеты позволяют легко размещать модели AnyLogic на веб-сайтах. В дополнение к Java-апплетам, AnyLogicProfessional поддерживает создание Java-приложений, в этом случае пользователь может запустить модель без инсталляции AnyLogic.

AnyLogic 7 включает в себя набор следующих библиотек:

1. **Библиотека моделирования процессов (ProcessModelingLibrary)**–эта библиотека является новой улучшенной версией старой Основной библиотеки. С новой

библиотекой не только проще строить процессные модели; в ней имеется больше параметров блоков, а также были разработаны новые библиотечные объекты. С Библиотекой моделирования процессов можно использовать специальные объекты разметки пространства, помогающие задавать местоположение агентов и ресурсов. Следует отметить, что Основная библиотека все еще включена в AnyLogic, и если диаграммы процесса в Вашей модели построены из блоков этой библиотеки, Вы сможете успешно запустить модель.

1. **Пешеходная библиотека (PedestrianLibrary)** –это высокоуровневая библиотека моделирования движения пешеходов в физическом пространстве. Она позволяет моделировать здания, в которых движутся пешеходы (станции метро, стадионы, музеи), а также улицы и другие места большого скопления людей. С помощью Пешеходной библиотеки Вы можете собирать статистику, эффективно визуализировать моделируемый процесс для валидации и представления Вашей модели. Вы можете собирать статистику плотности пешеходов в различных областях модели для того, чтобы убедиться, что сервисы смогут справиться с потенциальным ростом нагрузки, вычислить время пребывания пешеходов в каких-то определенных участках модели, выявить возможные проблемы, которые могут возникнуть при перепланировке интерьера здания, и т.д.
2. **Железнодорожная библиотека (RailLibrary)**– позволяетэффективно моделировать и визуализировать функционирование железнодорожных узлов и железнодорожных транспортных систем любого уровня сложности и масштаба. Сортировочные станции, пути погрузки/разгрузки больших предприятий, железнодорожные станции и вокзалы, станции метрополитена, шаттлы аэропортов, пути на контейнерных терминалах, движение трамваев и даже рельсовая транспортировка в угольных шахтах – все эти задачи могут быть легко и точно промоделированы с помощью Железнодорожной библиотеки.
3. **Библиотека моделирования потоков (TheFluidLibrary)**– позволяет Вам моделировать хранение и перенос жидкостей, объемного вещества или множества дискретных элементов, которые вы не хотите моделировать по отдельности.

# НАЗНАЧЕНИЕ ПАКЕТАANYLOGIC

Пакет AnyLogicпредназначен для создания имитационной модели существующей или проектируемой информационной системы.AnyLogic поддерживает все подходы к созданию имитационных моделей: процессно-ориентированный (дискретно- событийный), системно динамический и агентный, а также любую их комбинацию.

Уникальность, гибкость и мощность языка моделирования, предоставляемого AnyLogic, позволяет учесть любой аспект моделируемой системы с любым уровнем детализации. Графический интерфейс AnyLogic, инструменты и библиотеки позволяют быстро создавать модели для широко спектра задач от моделирования производства, логистики, бизнес-процессов до стратегических моделей развития компании и рынков.

# ОПИСАНИЕ ИНТЕРФЕЙСА ПРОГРАММЫ ANYLOGIC

После запуска программы AnyLogic на экране появится окно программы (Рис. 3.1).

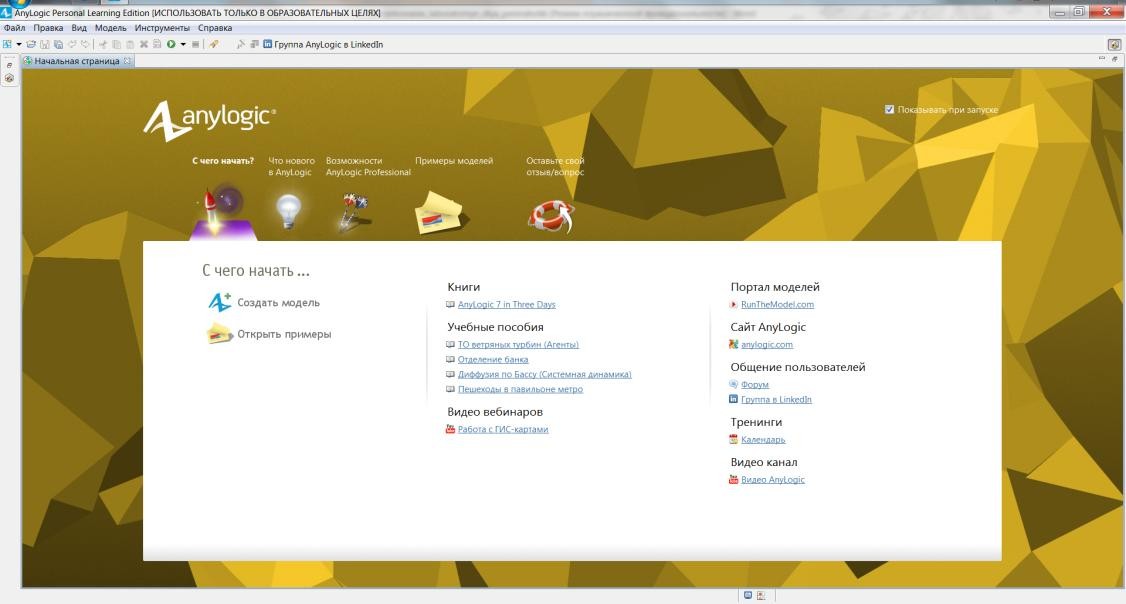


Рис. 3.1 Окно программы.

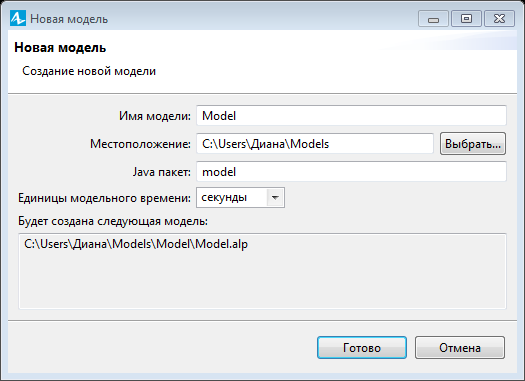
Для создания новогопроекта необходимо вызвать диалог Файл/Создать/Модельили нажать на соответствующий значок на панели инструментов, или нажать сочетание клавиш Ctrl+N. После этого возникнет диалоговое окно, в котором следует указать имя проекта и его расположение. В именах файловрекомендуетсяизбегать символы кириллицы, пробел или использовать для имени только цифры.

Рис. 3.2 Окно создания нового проекта.

На появившейся странице представлено поле для создания схемы модели из блоков.

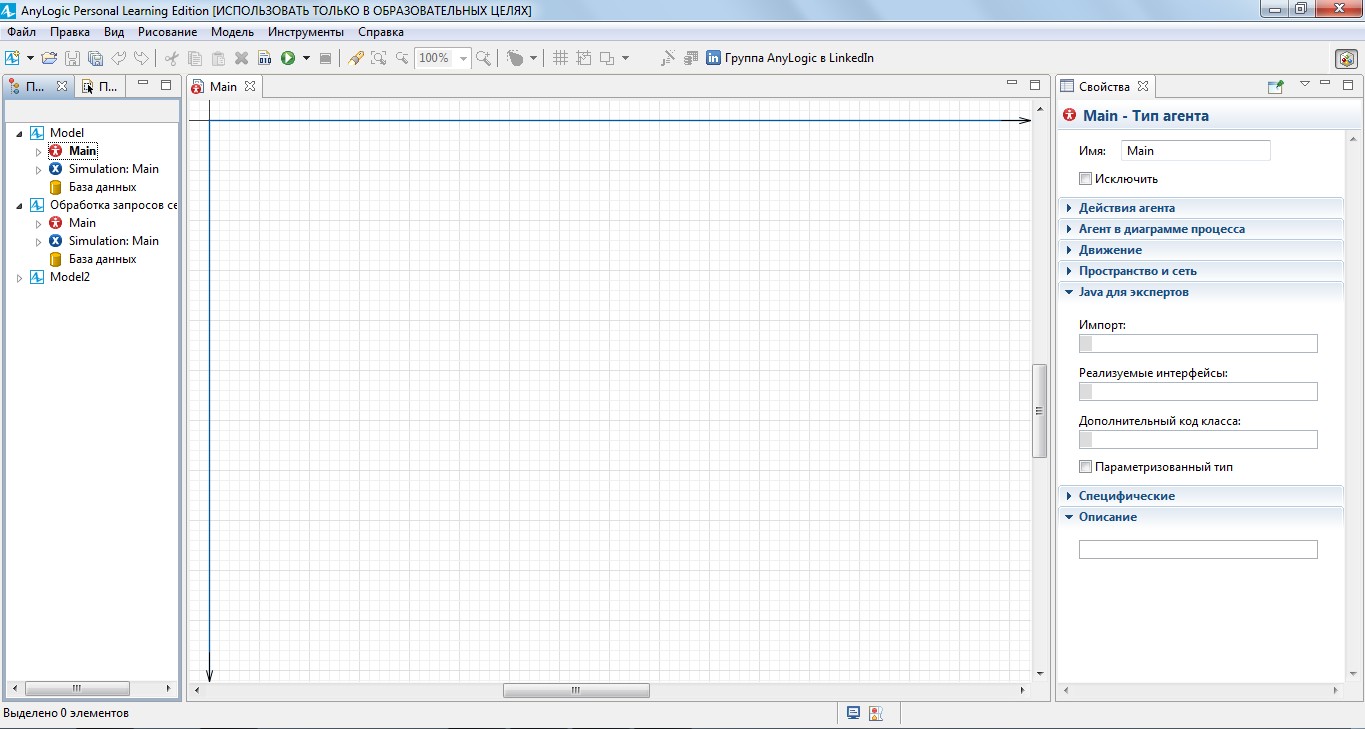


Рис. 3.3 Основное окно AnyLogic.

Основное окно программы содержит следующие части:

* Верхнее меню и панель.
* Дерево проекта и библиотеки элементов.
* Пространство для работы с моделью.
* Свойства модели или составляющих блоков.

Рассмотрим подробнее содержимое каждой из частей программы:

* Верхнее меню и панель открывают доступ ко всем возможным действиям с активной моделью и настройками самой программы, а также доступ к ряду отдельно вынесенных функций (открыть, сохранить, запуск модели и пр.)
* Дерево проекта отображает содержимое открытых проектов в виде иерархического дерева. Из него можно получить быстрый доступ к коду модели, анимации или схемам. В том числе в дереве отображаются созданные эксперименты с моделями.
* В библиотеках элементов предоставляется доступ ко всем стандартным элементам, из которых можно построить схему модели. Сюда не входят созданные пользователем элементы и элементы

анимации. При помощи соответствующих элементов возможно использование нескольких библиотек в рамках одной модели.

* В пространстве для работы с моделью могут располагаться анимации, схемы модели или составляющих ее элементов и код модели или элементов.
* В блоке свойств отображаются свойства выбранного пользователем элемента, или модели в целом. Помимо точной настройки каждого из элементов имеется возможность задания переменных, которые в дальнейшем могут использоваться в рамках всего проекта.

# ОСНОВНЫЕ СПОСОБЫ РАБОТЫ С МОДЕЛЬЮ

# Работа со схемой модели

Создание модели производится путем сборки ее из множества объектов, представленных в библиотеках, или созданных пользователем.

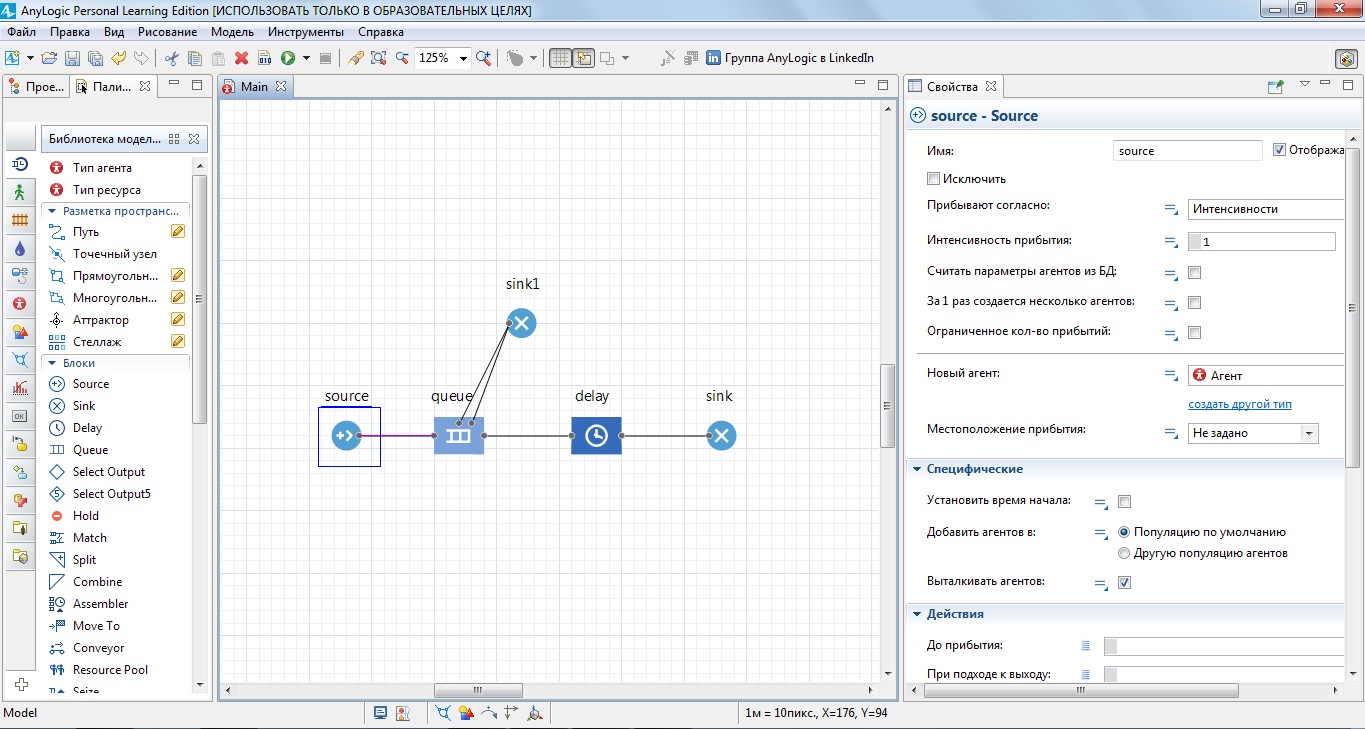
На рис. 4.1 представлен пример работы с моделью. В левом блоке активна Библиотека моделирования процессов, в основном блоке расположена схема модели, состоящая из 5 объектов и установленных между ними связей. В правом блоке выведены свойства объекта «source», который выделен в основном блоке (синяя рамка вокруг объекта).

Рис. 4.1 Работа со схемой модели.

Перенос объекта на схему осуществляется обычным перетаскиванием из коллекции библиотеки (drag-and-drop). Для установки связи между объектами необходимо дважды нажать левую кнопку мышина белом кружочке на периферии объекта, провести к другому объекту и нажать левую кнопку мыши на необходимом кружочке. Белые кружки на периферии являются портами входа и выхода объекта.

При построении блок-схем важно понимать, как происходит обмен заявками между активными объектами. Объекты библиотеки EnterpriseLibrary (Библиотеки моделирования процессов) соблюдают четкие правила передачи заявок. Заявки поступают в объект и покидают его через порты. Порт может работать только в одном направлении: или как входной, или как выходной (если не рассматривать передачу транспортеров и ресурсов в транспортной сети). Входной порт объекта может быть соединен только с выходным портом. Передача заявок выполняется согласно следующему протоколу:

Вначале объект, который намеревается передать заявку, посылает уведомление другому объекту.

Если принимающий объект соглашается принять заявку, то он пересылает запрос объекту, который намеревается передать заявку.

Заявка передается только в ответ на получение запроса. Если запрос прибывает тогда, когда заявка уже покинула передающий объект, то передается null.

Следовательно, заявка никогда не может прибыть в объект или покинуть его без предшествующего этому согласия другого объекта.

Более подробная информация по каждому из входов/выходов каждого из объектов, а так же по его свойствам может быть найдена в справочном руководстве AnyLogic.

В каждой из библиотек объекты делятся по своей функциональности на несколько групп:

# Библиотека моделирования процессов (ProcessModelingLibrary):

1. ***Разметка пространства*** (Путь, Точечный узел, Прямоугольный узел, Многоугольный узел, Аттрактор, Стеллаж)
2. ***Блоки***(Source, Sink, Delay, Queue, Select Output, Select Output5, Hold, Match, Split, Combain, Assembler, Move to, Conveyor, Resource Pool, Seize, Release, Service, Resource Send To, Resource Task

Start, Resource Task End, Resource Task, Расписание, Enter, Exit, Batch, Unbatch, Dropoff, Piclup, Restricted Area Start, Restricted Area End, Time Measure Start, Time Measure End, Resource Attach, Resource Detach, Rack System, Rack Store, Rack Pick, PWL Settings)

1. ***Дополнительныеблоки*** (Wait, SelectOutput In, SelectOutput Out, Plain Transfer)

# Пешеходная библиотека (PedestrianLibrary):

1. ***Разметка пространства*** (Стена, Прямоугольная стена, Округлая стена, Целевая линия, Сервис с очередями, Сервис с областью, Прямоугольная область, Многоугольная область, Аттрактор, Направление пеш.потока, Карта алотности пешеходов, Статистика потока)
2. ***Блоки*** (Ped Source, Ped Sinl, Ped Go To, Ped Service, Ped Wait, Ped Select Outpt, Ped Enter, Ped Exit, Ped Change Ground, Ped Area Description, Ped Group Assemble, Ped Group Chamge Formation, Ped Group Disassemble, Ped Settings)

# Железнодорожная библиотека (RailLibrary):

1. ***Разметка пространства*** (Ж/д путь, Точка ж/д пути)
2. ***Блоки*** (Train Source, Train Dispose, Train Move To, Train Couple, Train Decouple, Train Enter, Train Exit, Train Settings)

# Библиотека моделирования потоков (TheFluidLibrary):

1. ***Разметка пространства*** (Резервуар, Труба, Конвейер д/сыпучих материалов)
2. ***Блоки*** (Tank, Pipeline, Valve, Fluid Select Output, Fluid Select Input, Fluid Split, Fluid Merge, Bulk Conveyor, Fluid Convert, Fluid Exit, Fluid Enter, Agent To Fluid, Fluid To Agent)

# Работа с анимацией модели

Создание анимации в примере производится на основе элементов, размещенных в палитре «Разметка пространства» (рис. 4.2).

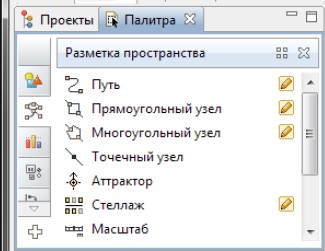


Рис. 4.2 Палитра «Разметка пространства».

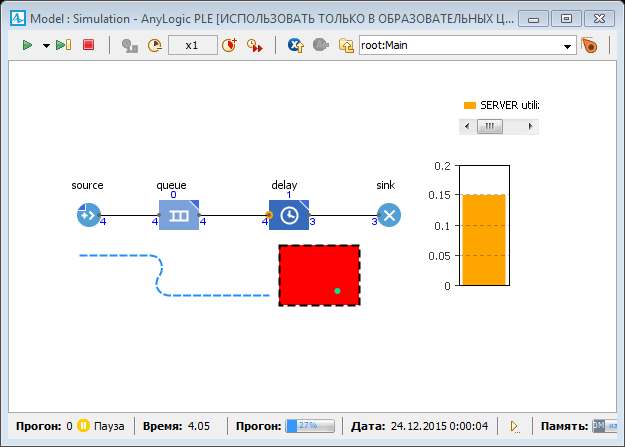
Пример простой анимации представлен на рис. 4.3, и он же во время процесса моделирования – на рис. 4.4.

Рис. 4.3 Пример анимации модели.

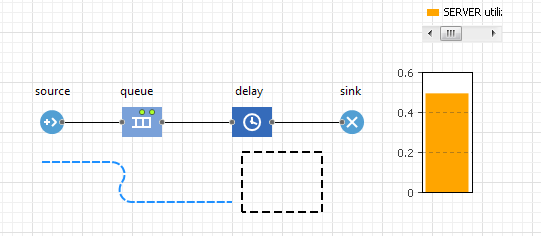


Рис. 4.4 Пример анимации модели во время моделирования.

Квадрат под элементом delayделает наглядным процесс работы этого элемента. В исходном состоянии, когда delayне обрабатывает никакую заявку, квадрат зеленого цвета; когда же delayначинает работать, квадрат меняет цвет накрасный, и количество точек в нем отображает количество обрабатываемых заявок.

Другой анимационный элемент – Столбиковая диаграмма. Это статистический элемент, отображающий статистику использования какого-либо объекта (в данном случае – delay).

# Реализация анимации

*Квадрат*

В палитре «Разметка пространства» найти элемент

«Прямоугольный узел». Основные настройки для узла показаны на рис. 4.5.

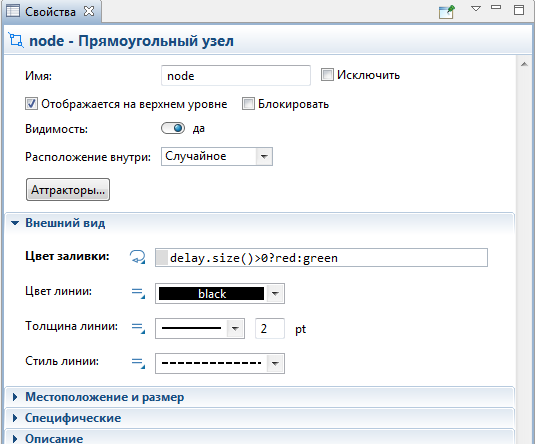


Рис. 4.5 Основные настройки элемента анимации «Квадрат».

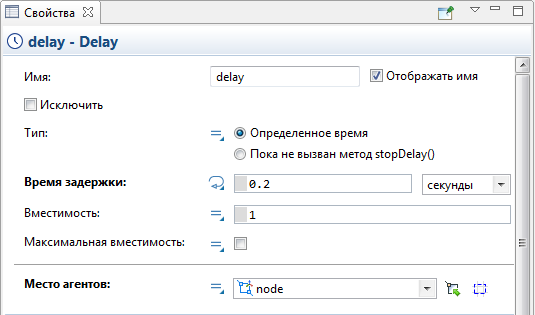
В настройках элемента delayнеобходимо в качестве Места агентов выбрать «node», как показано на рис. 4.6.

Рис. 4.6 Выбор необходимого Места агентов

*Столбиковая диаграмма*

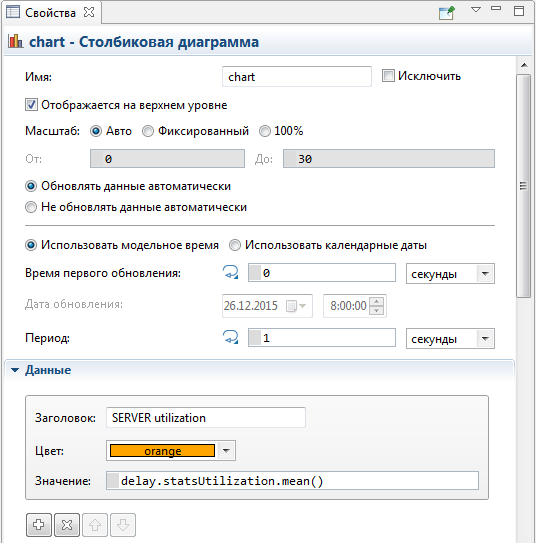
В палитре «Статистика» найти элемент «Столбиковая диаграмма». Основные настройки для этой диаграммы показаны на рис. 4.7.

Рис. 4.7 Необходимые настройки Столбиковой диаграммы Экспериментируя с настройками диаграммы вы также можете

менять ее положение с вертикального на горизонтальное, расположение поясняющей информации, задать точное местоположение и размер и др.

# Работа с кодом модели

Предположим, Вы создали в проекте некий Java-элемент, например, Java-класс (для создания этого элемента необходимо нажать правой кнопкой мыши на элемент модели верхнего уровня дерева и выбрать из контекстного меню**Создать/Java-класс**). Для перехода к коду созданного класса или объекта необходимо дважды кликнуть по нему левой кнопкой мыши в дереве проекта в левом

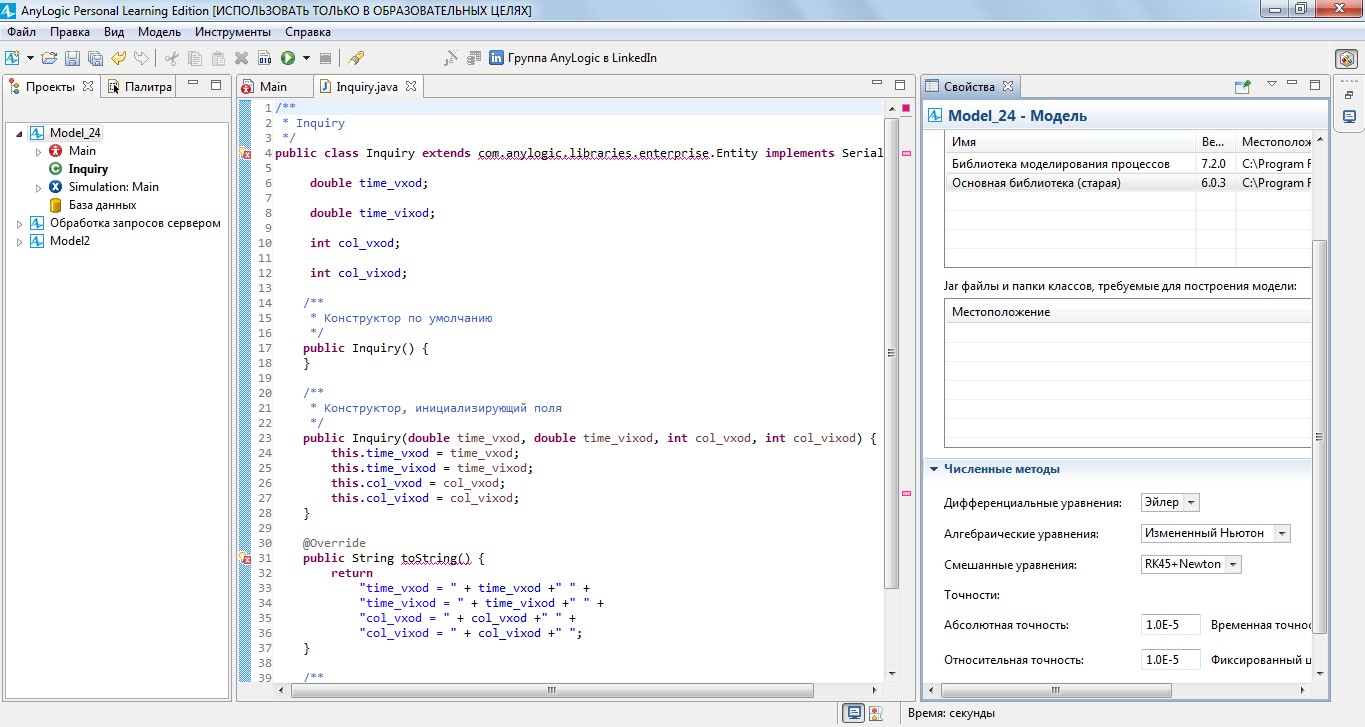
блоке. Пример окна редактирования кода модели представлен на рис.4.8.

Рис. 4.8 Пример работы с кодом модели.

И сам AnyLogic и вводимый в модели код базируются на языке программирования Java. Поддерживаются все конструкции языка, в том числе объявление функций и своих классов.

Взаимодействие кода и объектов модели осуществляется либо посредством прямого обращения к объектам, если это возможно, либо через переменные модели. Так же конструкции кода можно задавать в блоке свойств объектов в различных параметрах (условия сравнения, события при поступлении и уходе заявок и т.д.).

# ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

В лабораторной работе рассматривается задача разработки имитационной модели сети из NAT-сервера, трех клиентов и трех веб-служб.

Формулировка задачи:

Необходимо создать имитационную модель процесса работы участка сети из сетевого узла, трех клиентов и трех веб-служб, обменивающихся пакетами с данными. Создать анимацию к модели,

наглядно демонстрирующую поток пакетов и загруженность веб- служб.

Порядок выполнения работы:

Общий порядок разработки имитационной модели можно представить следующим образом:

* Разработка схемы модели.
* Разработка анимации модели.
* Проведение экспериментов.

Начнем построение модели с анализа необходимых библиотек. Для решения поставленной задачи вполне хватит объектов **Библиотеки Моделирования процессов (ProcessModelingLibrary)**, а именно элементов потока заявок, обработки и транспортировки по сети. Для задания необходимой веб-службы и клиента-отправителя нужно предусмотреть механизм хранения идентификатора отправителя и получателя в пакетах. Для этого можно создать собственный агент«Zapros», в котором разместить соответствующие переменные.

Каждый из клиентов может быть представлен набором элементов генерации заявки, отправки ее в сеть network и получением и уничтожением по окончании отработки (рис. 5.1).

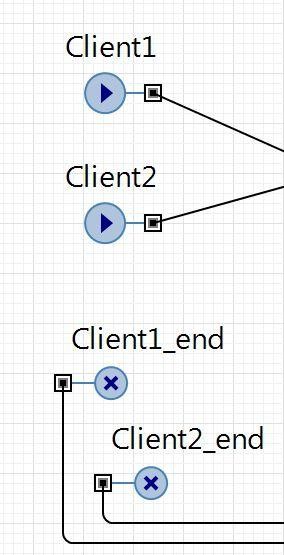


Рис. 5.1 Элементы, относящиеся к клиенту сети.

Создание нового агента выполняется с помощью пошаговой настройки параметров и дальнейшей настройки элемента вручную. Пошаговое создание и настройка агента показано на рис. 5.2-5.11.

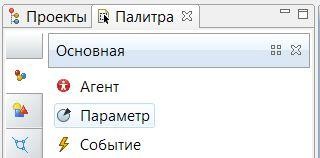


Рис. 5.2 Элемент «Агент» перетаскивается в основную область.

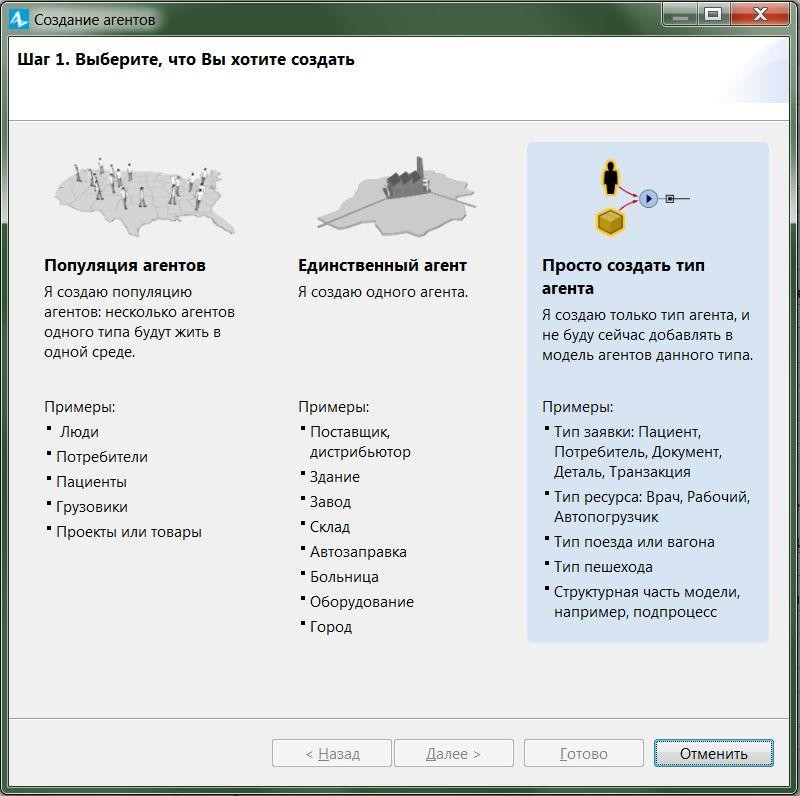


Рис. 5.3Создается обычный тип агента, который будет использоваться в дальнейшем в качестве заявки, отправляемой из элемента Source.

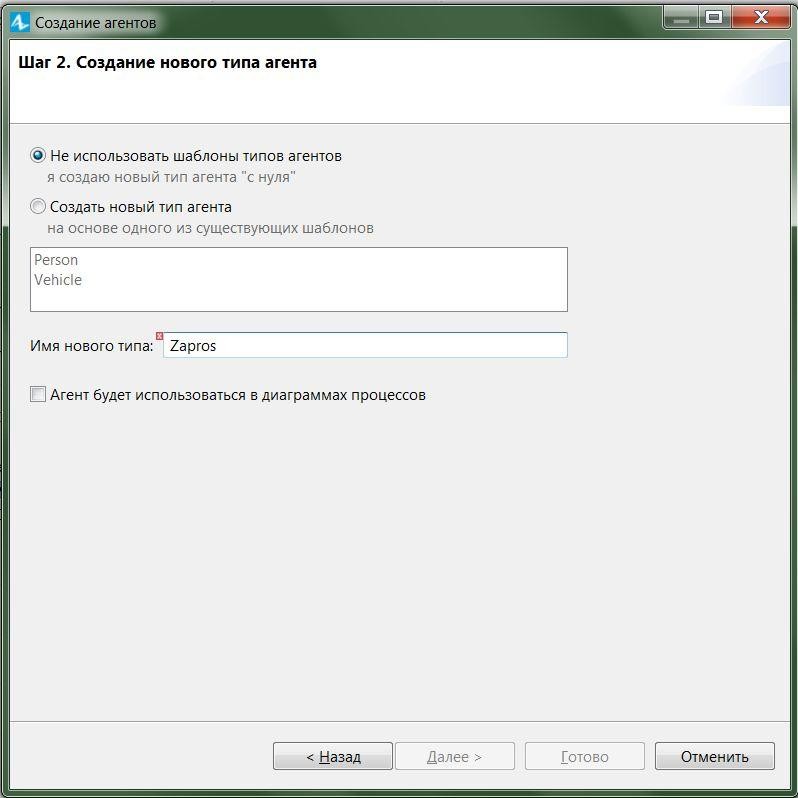


Рис. 5.4 Шаблоны для создания агента в этом случае использоваться не будут. Имя нового типа заполняется как Zapros.

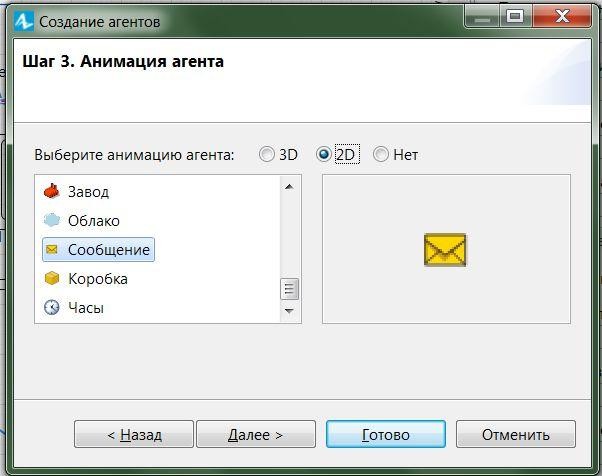


Рис. 5.5Для большего соответствия тематике, в качестве анимации будет использоваться символ «Сообщение».

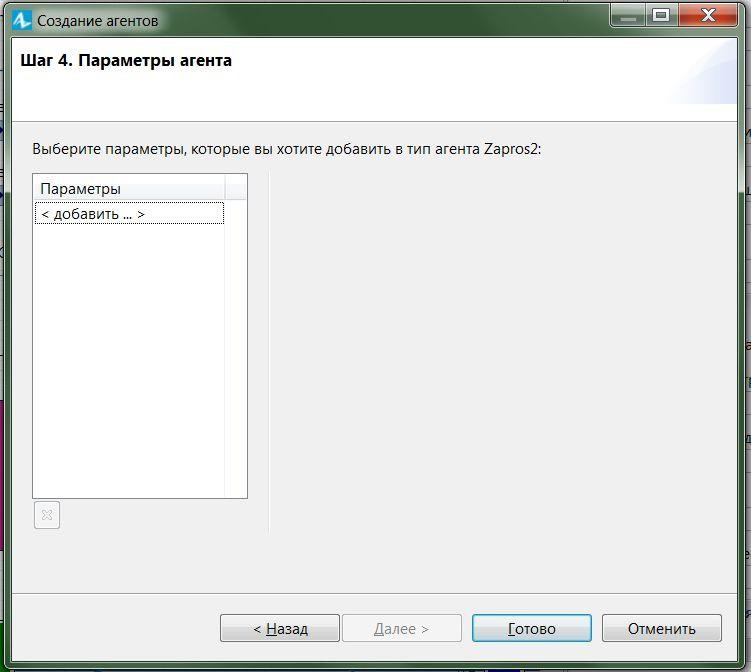


Рис. 5.6 Параметры не добавляются. В дальнейшем, в состав агента будут добавлены переменные. Переменными можно управлять извне класса агента.

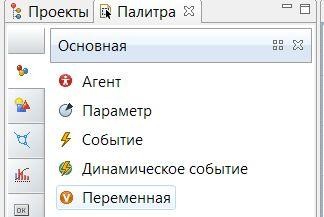


Рис. 5.7В новый агент добавляются три новые переменные.

Переменная to будет отвечать за сервер назначения сообщения, from отвечает за хранение номера клиента, от которого сообщение поступает, color\_Var задает цвет иконки сообщения для того, чтобы визуально различать сообщения, идущие с первого и второго клиентов.

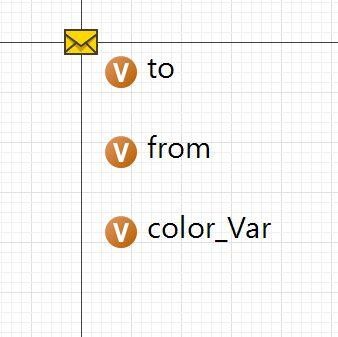


Рис. 5.8 Переменные агента «Zapros»

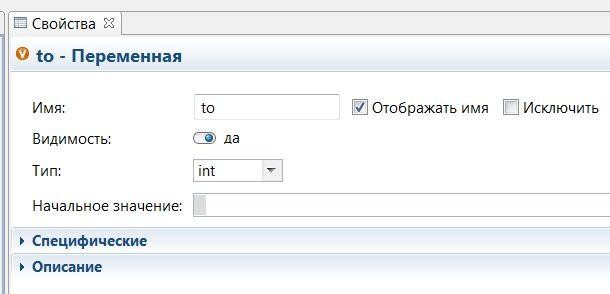


Рис. 5.9 Характеристики переменной «to».

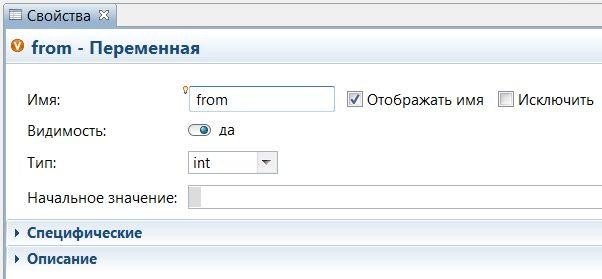


Рис. 5.10 Характеристики переменной «from».

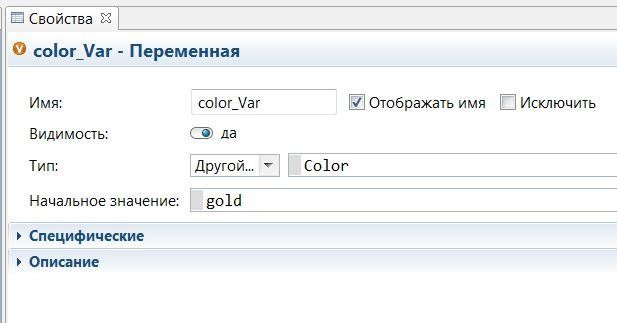


Рис. 5.11 Характеристики переменной «color\_Var»

В элементе генерации заявки необходимо указать тип заявки равным «Zapros». Так же нужно установить частоту генерации заявок, например экспоненциальную функцию распределения

«exponential(0.5)» и максимальное количество заявок, которые будут сгенерированы.

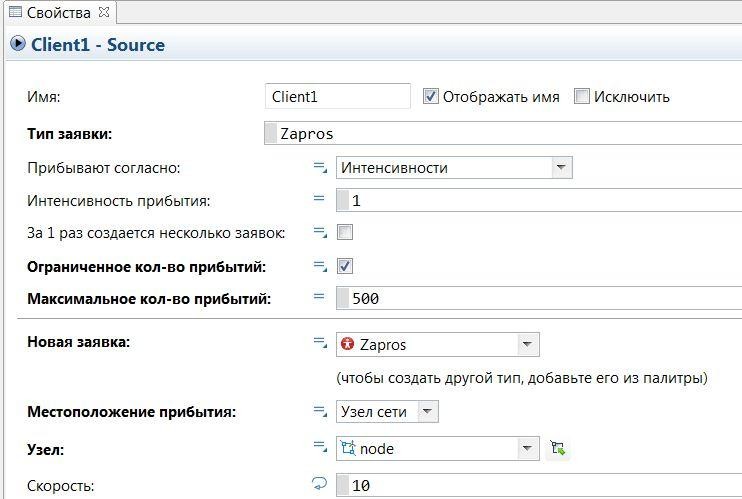


Рис. 5.12 Настройка элементов Source для использования агентов

«Zapros» в качестве заявок.

В разделе «Действия» нужно указать параметры генерации заявок (см. рис. 5.13):

entity.from=1;

entity.to=(**int**)round(uniform(0.5,2.4)); entity.color\_Var=red

Значения переменной color\_Var должны указываться разными для каждого клиента. Значение по умолчанию «gold», для первого сервера задано «red».

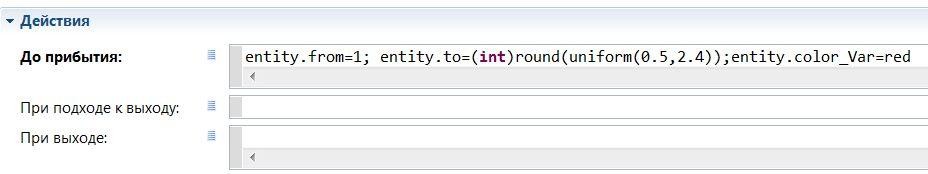


Рис. 5.13 Настройка действий элементов Source

Это позволит задать случайный конечный веб-сервис и указать идентификатор клиента-источника. Для 2-го и 3-его клиентов идентификаторы будут 2 и 3 соответственно.

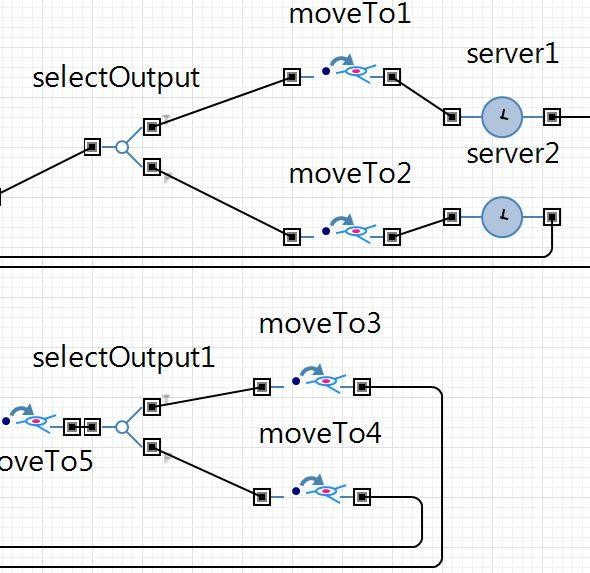
Часть сетевого узла и веб-служб может быть представлена схемой, изображенной на рис. 5.14.

Рис. 5.14 Схемасетевого узла и веб-служб.

Вобъектах«selectOutput»и«selectOutput1» происходит маршрутизация заявок посредством проверки значения поля «to» и

«from». Объекты «MoveTo*X*» отвечают за отображение перемещения заявок к веб-сервисам. Объекты «server*X*»соответствуют веб- сервисам. В этих объектах изменяются значения «время задержки»и

«вместимость», отвечающие за время обработки заявки и за вместимость.

После обработки заявок веб-службами остается отправить их обратно к клиентам. Это может быть реализовано по тому же

принципу, что и маршрутизация к веб-сервисам, только в данном случае необходимо проверять значение поля «from»заявок.

Так же на схеме модели располагаются объекты«Столбиковая диаграмм», которые служат для динамического отображения количества пакетов, которые отправлены клиентом, и на которые еще не получен ответ, и нагрузки на сервер. Изменяемые в нем свойства:

«Данные», отвечающие за количество запросов без ответа:

Client1.count()-Client1\_end.count()

И за количество объектов на сервере

server1.size()

Окончательный вид схемы модели представлен на рис.5.15.

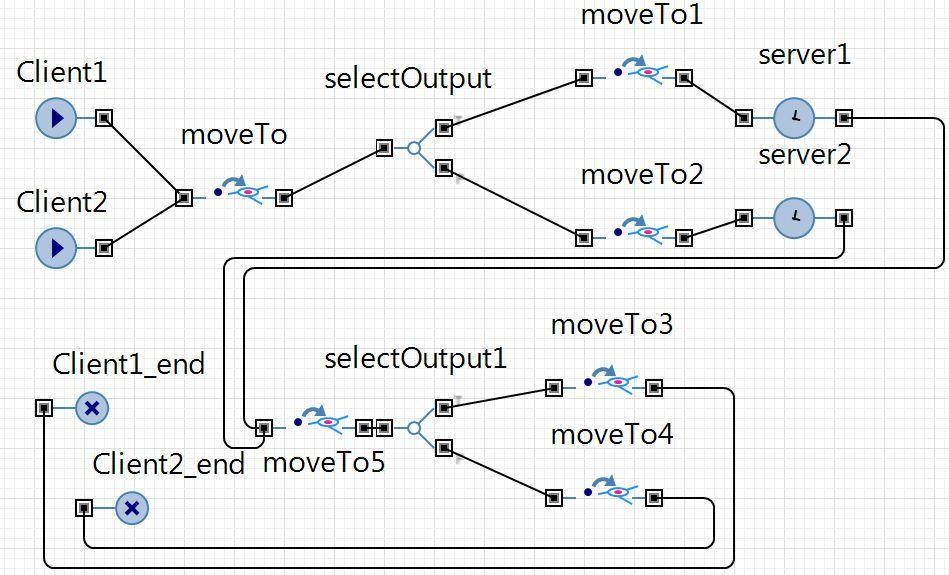


Рис. 5.15 Схема модели.

Анимация модели представлена на рис. 5.16. Все прямоугольные элементы и линии объединены в одну сеть.

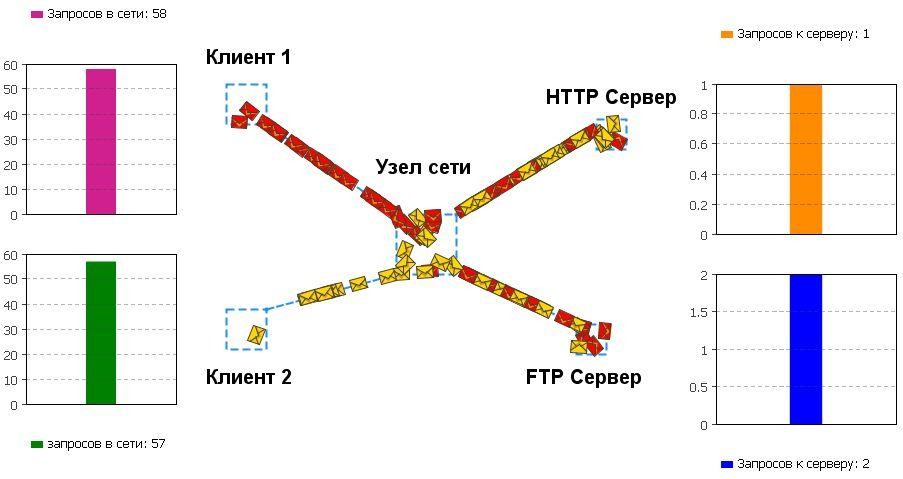


Рис. 5.16 Анимация модели.

# ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Пользуясь пакетом AnyLogic, по выбранной студентом предметной области разработать имитационную модель системы, указанную в задании. Имитационная модель должна иметь анимацию и набор управляющих элементов для изменения характеристик модели. Провести ряд экспериментов с моделью сформировать статистику.

Варианты заданий:

# Моделирование участка сети.

Создать модель участка сети, в которой не менее 3-х клиентов передают случайным образом сообщения для 3-х и более серверов разного назначения. Пакеты задерживаются на сервере на определенное время (задается назначением сервера). В модели должна быть представлена анимация и графики.

# Линейная регрессионная модель.

Построить регрессионную модель по заданным точкам, вычислить среднюю ошибку аппроксимации и построить линии ошибки как сверху, так и снизу линии регрессии. Построить новый

график, выбросив точки, расстояние от которых до линии регрессии больше, чем средняя ошибка и построить новую гипотезу и новый график.

# Моделирование поведения пешеходов на примере магазина самообслуживания.

Построить модель магазина с 5-ю прилавками и 2 и более кассами. Покупатели случайным образом подходят к прилавкам и случайным образом берут 1-2 товара. При подходе к кассам у покупателя должно быть не менее 7 товаров. Если в кассы очередь более 5 человек на одну кассу, то покупатель с вероятностью 20-30% выходит из магазина, оставляя покупки в магазине, иначе встает в кассу с минимальной длиной очереди. Встав в кассу, покупатель обслуживается по времени пропорциональному количеству покупок, которые он имеет при себе (по 5 секунд на каждую покупку+20 секунд). На анимации должны отображаться графики, содержащие процентное соотношение что-то купивших и ушедших без покупок посетителей. Для каждой кассы должна отображаться длина очереди и должно отображаться средне количество товаров, купленных в магазине одним покупателем.

# КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите основное назначение элементов библиотеки EnterpriseLibrary.
2. Опишите основное назначение элементов библиотеки PedestrianLibrary.
3. Опишите основное назначение элементов библиотеки AgentBasedLibrary.
4. Опишите основное назначение элементов библиотеки BusinessGraphicsLibrary.
5. Опишите механизм продвижения заявок по схеме модели.
6. Дайте описание функций распределения uniform иtriangular.
7. Как могут быть заданы и задействованы пользовательские переменные в модели?
8. Приведите пример сети на основе элементов группы

«network» библиотеки EnterpriseLibrary.

1. Приведите пример построения графика на основе элементов библиотеки BusinessGraphicsLibrary.
2. Приведите пример работы с сущностями на основе библиотеки PedestrianLibrary.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

* 1. Горбунов А.В. Моделирование систем [Текст]. В 2 ч. Ч. 1. Основы теории моделирования систем: Учебное пособие / А.В.Горбунов. – Таганрог, 2007. – 57 с.
  2. Питерсон Д. Теория сетей Петри и моделирование систем = Petrinettheoryandthemodelingofsystems [Текст] / Дж. Питерсон ; пер. с англ. М.В. Горбатовой [и др.] ; под ред. В.А. Горбатова. – М. : Мир, 1984. – 264 с. : ил. – Парал. тит. л.: англ. – Библиогр.: с. 234-261. – 8 400 экз.. – (в пер.)
  3. AnilK. Jain, JianchangMao, K.M. Mohiuddin. Artificial Neural Networks: A Tutorial [Текст]/Computer, Vol.29, No.3, March 1996, pp. 31-44.
  4. Топорков В.В. Модели распределенных вычислений [Текст] / В.В. Топорков. – М. :Физматлит, 2004. – 315 с. : ил. – Библиогр.: с. 299-308. – ISBN 5-9221-0495-0 (в пер.)
  5. Мартин Д. Системный анализ передачи данных : [в 2 т.] [Текст] = Systems analysis for data transmission : пер. сангл. / Дж. Мартин . – М. : Мир, 1975. – Парал. тит. л.: англ. Т. 2 : Проектирование систем передачи данных / под ред. В.С. Лапина. – 431 с. : ил., табл. – (в пер.)
  6. Советов Борис Яковлевич Моделирование систем: Учебник для вузов [Текст] / Б. Я. Советов, С. А. Яковлев. – М.: Высш. шк., 2009. – 343 с.: ил. – Библиогр.: с. 340-341 (54 назв.) ISBN 978-5-06-006173-4
  7. Муха В. С. Вычислительные методы и компьютерная алгебра: учеб.-метод. пособие. [Текст]/ М.С.Муха – 2-е изд., испр. и доп. – Минск: БГУИР, 2010.- 148 с.: ил, ISBN 978-985-488-522-3
  8. Нейронные сети : полный курс = Neuralnetworks : a comprehensivefoundation [Текст] / С. Хайкин. – Изд. 2-е, испр. – М. : Вильямс, 2006. – 1103 с. : ил. – Парал. тит. л.: англ. – Доп. тираж 2000 экз.. – ISBN 5-8459-0890-6 (в пер.)