Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ И ДИЗАЙНА»**

Кафедра информационных технологий

Моделирование экономических процессов и систем

Методические указания к выполнению
контрольной работы
для направления 09.03.03 «Прикладная информатика»
профиля «Прикладная информатика в экономике»
заочной формы обучения

Составитель: Т. А. Кравец

Санкт-Петербург
2016

Содержание

Введение

Моделирование – метод решения задач, при использовании которого исследуемая система заменяется более простым объектом, описывающим реальную систему и называемым моделью.

Моделирование применяется в случаях, когда проведение экспериментов над реальной системой невозможно или нецелесообразно, например, из-за высокой стоимости или длительности проведения эксперимента в реальном масштабе времени.

Имитационная модель позволяет получать подробную статистику о различных аспектах функционирования системы в зависимости от входных данных.

Целью моделирования в конечном счете является изучение поведения системы при различных условиях и принятие обоснованного, целесообразного управленческого решения.

Современный программный продукт AnyLogic является системой моделирования, которая поддерживает весь арсенал новейших информационных технологий, включая развитые графические оболочки для целей конструирования моделей и интерпретации выходных результатов моделирования, мультимедийные средства, анимацию в реальном масштабе времени, объектно-ориентированное программирование, internet решения и др. Разработчиком продукта является компания «Экс Джей Текнолоджис» (XJ Technologies), Санкт-Петербург.

Разработка модели выполняется в графическом редакторе AnyLogic с использованием многочисленных средств поддержки, упрощающих работу. Построенная модель затем компилируется встроенным компилятором AnyLogic и запускается на выполнение. В процессе выполнения модели пользователь может наблюдать ее поведение, изменять параметры модели, выводить результаты моделирования в различных формах и выполнять разного рода компьютерные эксперименты с моделью.

Для реализации специальных вычислений и описания логики поведения объектов AnyLogic позволяет использовать мощный современный язык Java.

Основными строительными блоками модели AnyLogic являются *активные объекты*, которые позволяют моделировать любые объекты реального мира. Чтобы создать модель AnyLogic, необходимо создать классы активных объектов (или использовать объекты библиотек AnyLogic) и задать их взаимосвязи. AnyLogic интерпретирует создаваемые графическе классы активных объектов в классы *Java*, поэтому можно пользоваться всеми

преимуществами объектно-ориентированного моделирования.

В свою очередь активные объекты могут содержать вложенные объекты, причем уровень вложенности не ограничен. Это позволяет производить декомпозицию модели на любое количество уровней детализации.

Активные объекты могут быть повторно используемыми. Создав класс активного объекта, можно создать любое количество объектов – экземпляров данного класса.

Каждый активный объект имеет структуру (совокупность включенных в него активных объектов и их связи), а также поведение, определяемое совокупностью переменных, параметров, стейтчартов и т. п. Каждый экземпляр активного объекта в работающей модели имеет свое собственное поведение, он может иметь свои значения параметров, функционирует независимо от других объектов, взаимодействуя с ними и с внешней средой.

При построении модели используются средства визуальной разработки (введения состояний и переходов стейтчарта, введения пиктограмм переменных и т. п.), задания численных значений параметров, аналитических записей соотношений переменных и аналитических записей условий наступления событий. Основной технологией программирования в AnyLogic является визуальное программирование – построение с помощью графических объектов и пиктограмм иерархий структуры и поведения активных объектов. AnyLogic является надстройкой над языком *Java* – одним из самых мощных и в то же время самых простых современных объектно-ориентированных языков. Все объекты, определенные пользователем при разработке модели с помощью графического редактора, компилируются в конструкции языка *Java*, а затем происходит компиляция всей собранной программы на *Java*, задающей модель, в исполняемый код. Хотя программирование сведено к минимуму, разработчику модели необходимо иметь некоторое представление об этом языке (например, знать синтаксически правильные конструкции).

Основными средствами описания поведения объектов являются переменные, события и диаграммы состояний. *Переменные* отражают изменяющиеся характеристики объекта. *События* могут наступать с заданным интервалом времени и выполнять заданное действие. *Диаграммы состояний (или стейтчарты)* позволяют визуально представить поведение объекта во времени под воздействием событий или условий, они состоят из графического изображения состояний и переходов между ними (т. е. по сути это конечный автомат). Любая сложная логика поведения объектов модели может быть выражена с помощью комбинации стейтчартов,

дифференциальных и алгебраических уравнений, переменных, таймеров и программного кода на *Java*. Алгебраические и дифференциальные уравнения записываются аналитически.

Важным понятием в системах имитационного моделирования является модельное временя. *Модельное время* – это условное логическое время, в единицах которого определено поведение всех объектов модели. В моделях AnyLogic модельное время может изменяться либо непрерывно, если поведение объектов описывается дифференциальными уравнениями, либо дискретно, переключаясь от момента наступления одного события к моменту наступления следующего события, если в модели присутствуют только дискретные события. Моменты наступления всех планируемых событий в дискретной модели исполнительная система хранит в так называемом календаре событий, выбирая оттуда наиболее раннее событие для выполнения связанных с ним действий. Значение текущего времени в моделях AnyLogic может быть получено с помощью функции *time()*.

Единицу модельного времени разработчик модели может интерпретировать как любой отрезок времени: секунду, минуту, час или год. Важно только, чтобы все процессы, зависящие от времени, были выражены в одних и тех же единицах.

AnyLogic имеет удобные средства представления функционирования моделируемой системы в живой форме динамической анимации, что позволяет «увидеть» поведение сложной системы. Визуализация процесса функционирования моделируемой системы позволяет проверить адекватность модели, выявить ошибки при задании логики.

Средства анимации позволяют пользователю легко создавать виртуальный мир (совокупность графических образов, ожившую мнемосхему), управляемый динамическими параметрами модели по законам, определенным пользователем с помощью уравнений и логики моделируемых объектов. Графические элементы, добавленные на анимацию, называются динамическими, поскольку все их параметры: видимость, цвет и т. п. – можно сделать зависимыми от переменных и параметров модели, которые меняются со временем при выполнении модели.

С помощью совершенной технологии визуализации работающих моделей AnyLogic можно создавать интерактивные анимации произвольной сложности, связывая графические объекты (в т. ч. импортированные чертежи) во встроенном редакторе с объектами модели. Как и модель, анимация имеет иерархическую структуру, которая может динамически изменяться. Возможно создание нескольких точек зрения или нескольких

уровней детальности в пределах одной анимации. Элементы управления и развитая бизнес-графика превращают анимацию модели в настоящую панель управления для оценки эффективности решений. В AnyLogic поддерживается как двумерная, так и трёхмерная анимация.

Многие системы моделирования позволяют менять параметры модели только до запуска модели на выполнение. AnyLogic позволяет пользователю вмешиваться в работу модели, изменяя параметры модели в процессе ее функционирования. Примером таких средств являются *слайдеры*, которые могут быть введены в окно анимации.

Настоящие методические указания предназначены для изучения дисциплины «Моделирование экономических процессов и систем» студентами заочной формы обучения направления 09.03.03 «Прикладная информатика» профиля подготовки «Прикладная информатика в экономике» и выполнения контрольной работы.

В контрольной работе излагается материал по разработке дискретно-событийной модели банковского отделения.

Для выполнения контрольной работы необходимо изучить теоретический материал, приведенный в методических указаниях и разобрать пример. По выбранному варианту индивидуального задания выполнить контрольную работу средствами Anylogic 7 Professional 7.0.2 и оформить ее в соответствии с требованиями, приведенными в настоящих методических указаниях.

Для успешного изучения и освоения материала в настоящих методических указаниях приводится список литературы, который так же поможет подготовиться к экзамену.

Готовая контрольная работа передается для проверки на кафедру в соответствии с установленными заочным отделом правилами передачи контрольных работ. Для проверки преподавателю необходимо предоставить модель (выполняется в Anylogic Professional 7.0.2).

1. Требования к выполнению контрольной работы. Выбор варианта

Контрольная работа по дисциплине «Моделирование экономических процессов и систем» выполняется студентами направления 09.03.03 заочной формы обучения.

По результатам изученного материала, представленного в настоящих учебно-методических указаниях, студент выполняет расчеты по индивидуальному заданию. Вариант исходных данных для самостоятельного выполнения задания выбирается в соответствии с последними цифрами номера зачетки из табл. 1. Внутри одной студенческой группы не может быть выбран и выполнен один и тот же вариант контрольной работы.

В контрольной работе необходимо построить имитационную модель банковского отделения средствами **Anylogic 7 Professional 7.0.2** и провести расчеты по индивидуальному заданию. **Важно**, проводить разработку модели средствами пакета **версии 7 .0.2**, которая установлена на кафедре.

Готовая работа передается для проверки на кафедру в соответствии с установленными заочным отделом правилами передачи контрольных работ. Для проверки преподавателю необходимо предоставить модель (выполняется в Anylogic Professional 7.0.2).

2. Варианты индивидуальных заданий

В AnyLogic включено 37 генераторов случайных величин с наиболее часто встречающимися вероятностными распределениями: равномерным, экспоненциальным, Бернулли, биномиальным и т.п. Их описание можно найти в руководстве пользователя AnyLogic.

Вызвать метод очень просто, например, exponential(0.6) или uniform(-1,1), который вернет соответствующее случайное значение.

За детальным описанием функций и их параметров обращайтесь к руководству пользователя или справочнику классов (см. методы класса Func). Для вызова руководства пользователя, справочника классов AnyLogic выберите соответствующие пункты меню Справка.

Внесите изменения в созданную модель банковского отделения согласно варианту индивидуального задания (*табл. 1*), запустите модель и изучите ее поведение.

Таблица 1 **Варианты индивидуальных заданий**

Вари-ант	Распределение вероятности прихода клиентов в банк	Вероятность обращения к кассиру/к	Время обслуживания клиента кассиром	Количество кассиров
1	Экспоненциальное	1/1	4 ± 2	1
2	Экспоненциальное	1/2	6 ± 2	2
3	Экспоненциальное	2/1	8 ± 2	3
4	Экспоненциальное	1/3	7 ± 2	4
5	Экспоненциальное	3/1	9 ± 2	5
6	Нормальное	1/1	8 ± 2	1
7	Нормальное	1/2	7 ± 2	2
8	Нормальное	2/1	9 ± 2	3
9	Нормальное	1/3	4 ± 2	4
10	Нормальное	3/1	6 ± 2	5
11	Треугольное	1/1	2 ± 2	1
12	Треугольное	1/2	7 ± 2	2
13	Треугольное	2/1	9 ± 2	3
14	Треугольное	1/3	4 ± 2	4
15	Треугольное	3/1	6 ± 2	5
16	Равномерное	1/1	4 ± 2	1
17	Равномерное	1/2	6 ± 2	2
18	Равномерное	2/1	7 ± 2	3
19	Равномерное	1/3	8 ± 2	4
20	Равномерное	3/1	9 ± 2	5

Проанализируйте поведение модели. Постройте графики и диаграммы переменных, характеризующих поведение модели, поместите их в отчет.

По результатам работы необходимо оформить отчет с выводами.

3. Создание модели банковского отделения

Библиотека Моделирования Процессов AnyLogic подерживает дискретно-событийный, или, если быть более точным, "процессный" подход моделирования. С помощью объектов Библиотеки Моделирования Процессов Вы можете моделировать системы реального мира, динамика которых представляется как последовательность операций (прибытие, задержка, захват ресурса, разделение, ...) над некими сущностями (entities, по-русски - транзакты, заявки), представляющими клиентов, документы, звонки, пакеты данных, транспортные средства и т.п. Эти сущности пассивны, они сами не контролируют свою динамику, но могут обладать определёнными атрибутами, влияющими на процесс их обработки (например, тип звонка, сложность работы) или накапливающими статистику (общее время ожидания, стоимость).

Потоковые диаграммы AnyLogic иерархичны, масштабируемы, расширяемы и объектно-ориентированы, что позволяет пользователю моделировать сложные системы любого уровня детальности. Другой важной особенностью Библиотеки моделирования процессов является возможность создания достаточно сложных анимаций процессных моделей.

В этом разделе учебного пособия мы создадим модель простой системы обслуживания, а именно модель банковского отделения. В банковском отделении находятся банкомат и стойки банковских кассиров, что позволяет быстро и эффективно обслуживать посетителей банка. Операции с наличностью клиенты банка производят с помощью банкомата, а более сложные операции, такие как оплата счетов – с помощью кассиров.

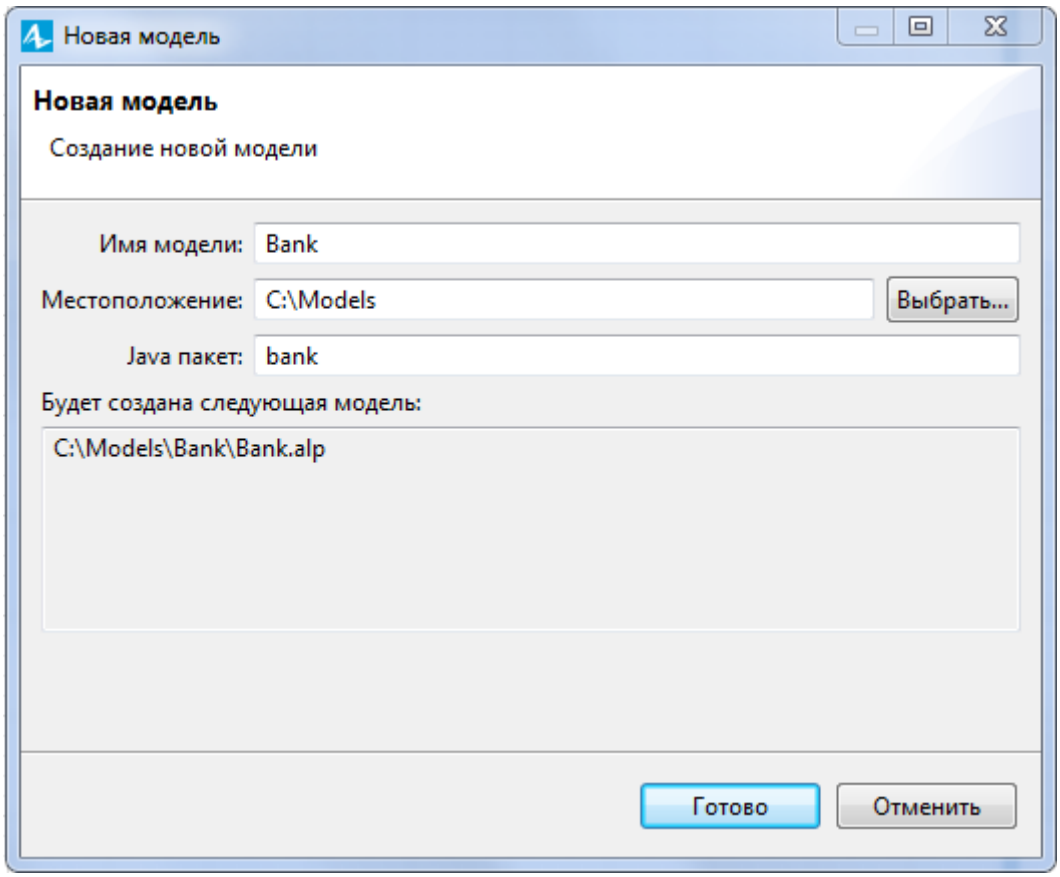
Обратите внимание, что для каждого шага моделирования имеются контрольные модели. Вы можете использовать их, чтобы сравнить свою модель с контрольной. Далее приведена ссылка на контрольную модель для этого шага. Ссылки на контрольные модели находятся в конце описания некоторых шагов. Просто щелкните ссылку, чтобы открыть модель.

3.1. Создание новой модели

Вначале мы создадим простейшую модель, в которой будем рассматривать обслуживание людей банкоматом.

Создайте новую модель

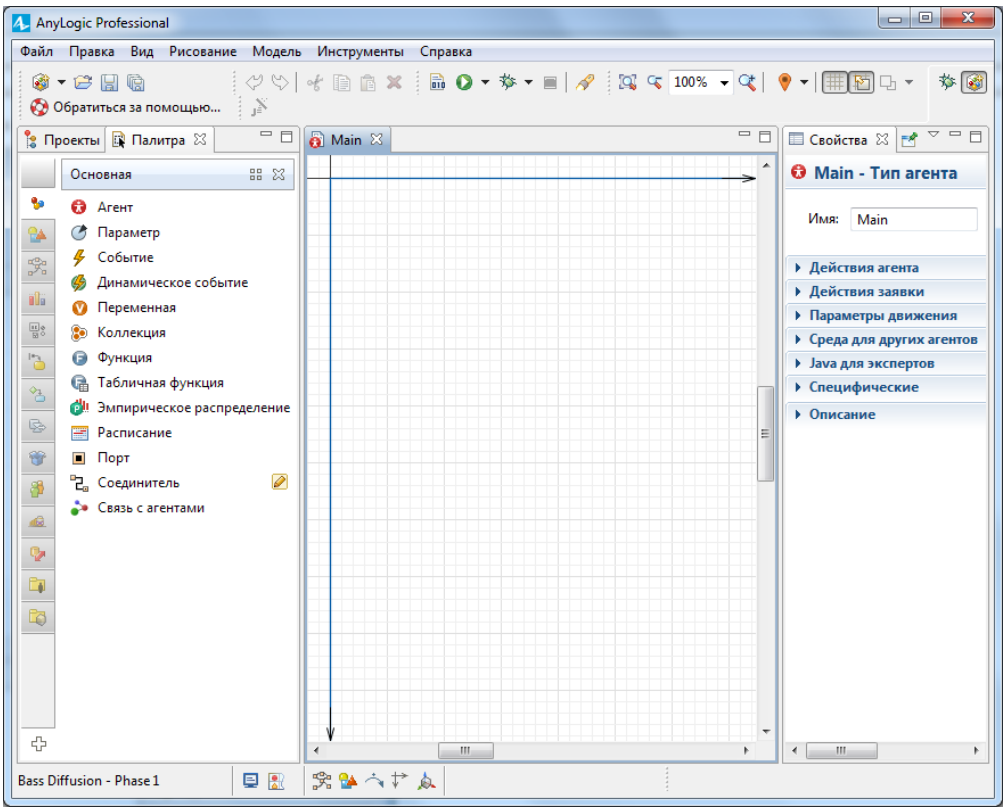
1. Щелкните мышью по кнопке панели инструментов **Создать** .
   Появится диалоговое окно **Новая модель.**
2. Задайте имя новой модели. В поле **Имя модели** введите Bank.



3. Выберите каталог, в котором будут сохранены файлы модели. Если Вы хотите сменить предложенный по умолчанию каталог на какой-то другой, Вы можете ввести путь к нему в поле **Местоположение** или выбрать этот каталог с помощью диалога навигации по файловой системе, открывающегося по нажатию на кнопку **Выбрать.**
4. Щелкните мышью по кнопке **Готово**, чтобы завершить процесс.

Вы создали новую модель. В ней уже имеется один тип агента Main и эксперимент Simulation. Агенты - это главные строительные блоки модели AnyLogic. В нашем случае агент Main послужит местом, где мы зададим всю логику модели: здесь мы расположим чертеж банковского отделения и зададим диаграмму процесса потока клиентов.

В центре рабочей области находится графический редактор диаграммы типа агента Main.



В левой части рабочей области находятся панель **Проекты** и панель **Палитра**. Панель **Проекты** обеспечивает легкую навигацию по элементам моделей, открытых в текущий момент времени. Поскольку модель организована иерархически, то она отображается в виде дерева. Панель **Палитра** содержит разделенные по палитрам элементы, которые могут быть добавлены на диаграмму типа агента или эксперимента.

В правой рабочей области будет отображаться панель **Свойства.** Панель **Свойства** используется для просмотра и изменения свойств выбранного в данный момент элемента (или элементов) модели. Когда вы выделяете какой-либо элемент, например, в панели **Проекты** или графическом редакторе, панель **Свойства** показывает свойства выбранного элемента.

Теперь мы можем настроить нашу модель, созданную с помощью **Мастера создания модели**.

Создание диаграммы процесса

Теперь мы зададим динамику процесса, создав диаграмму из блоков **Библиотеки моделирования процессов.**

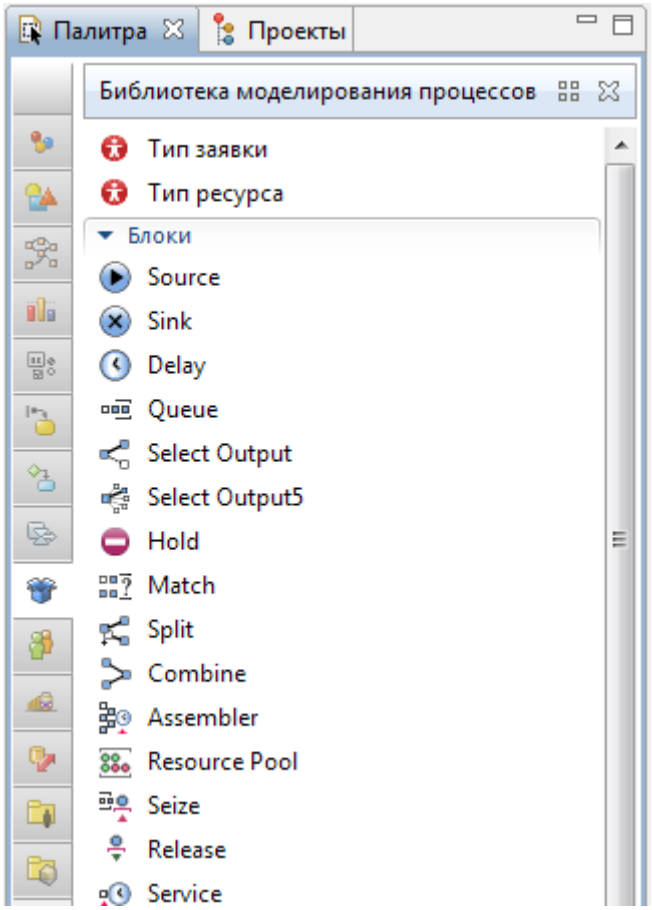
Каждый блок задает определенную операцию, которая будет производиться над проходящими по диаграмме процесса заявками.

Диаграмма процесса в AnyLogic создается путем добавления объектов библиотеки из палитры на диаграмму агента, соединения их портов и

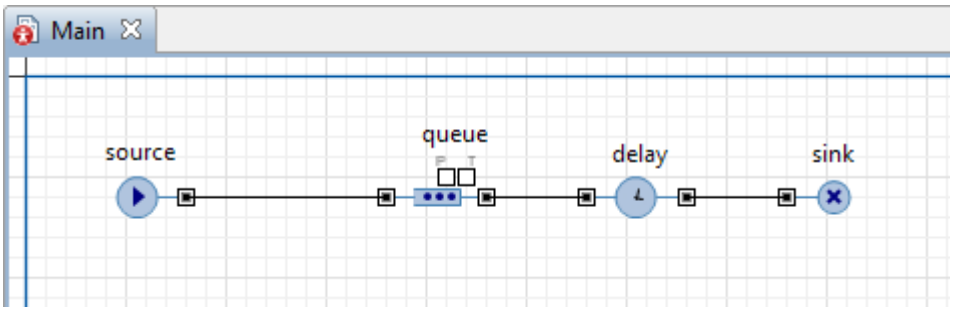
изменения значений свойств блоков в соответствии с требованиями Вашей модели.

Создайте диаграмму процесса

1. Вначале откройте палитру **Библиотека моделирования процессов** панели **Палитра**, щелкнув мышью по соответствующей иконке на вертикальной панели слева от палитры:



2. Добавьте блоки **Библиотеки моделирования процессов** на диаграмму и соедините их, как показано на рисунке. Чтобы добавить объект на диаграмму, перетащите требуемый элемент из палитры в графический редактор.



3. Пока вы перетаскиваете блоки и располагаете их рядом друг с другом, Вы можете видеть, как появляются соединительные линии между блоками. Будьте внимательны, эти линии должны соединять только порты, находящиеся с правой или левой стороны иконок.

Данная схема моделирует простейшую систему очереди, состоящую из источника заявок, задержки (и очереди перед задержкой) и финального уничтожения заявок.

Скажем пару слов об этих объектах диаграммы.

Объект **Source** генерирует заявки определенного типа. Обычно он используется в качестве начальной точки диаграммы процесса, формализующей поток заявок. В нашем примере заявками будут посетители банка, а объект **Source** будет моделировать их приход в банковское отделение.

Объект **Queue** моделирует очередь заявок, ожидающих приема объектами, следующими за данным в диаграмме процесса. В нашем случае он будет моделировать очередь клиентов, ждущих освобождения банкомата.

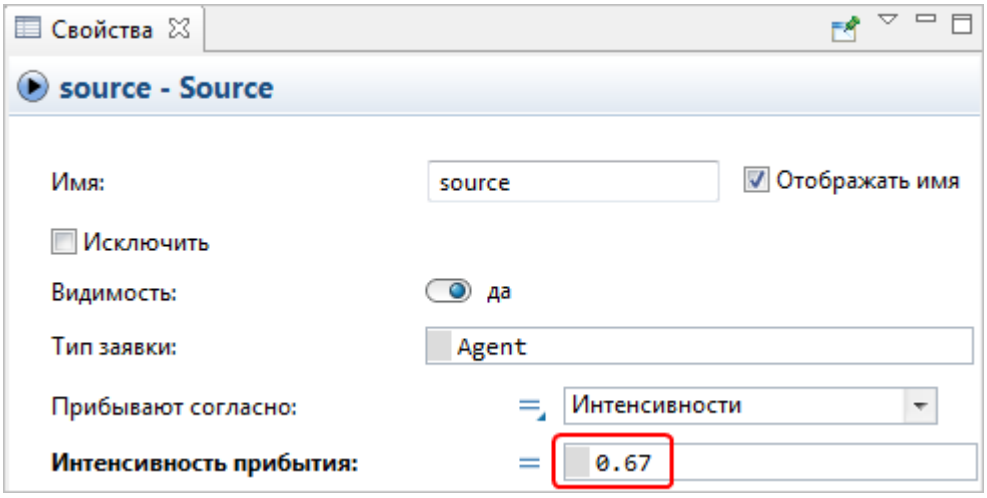
Объект **Delay** задерживает заявки на заданный период времени, представля в нашей модели банкомат, у которого посетитель банковского отделения тратит свое время на проведение необходимой ему операции.

Объект **Sink** уничтожает поступившие заявки. Обычно он используется в качестве конечной точки потока заявок (и диаграммы процесса соответственно).

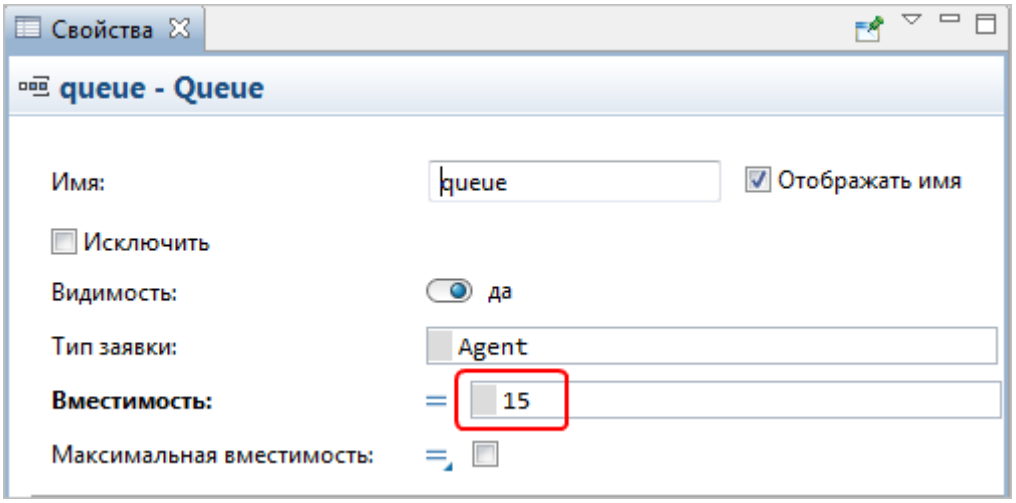
За детальным описанием объектов **Библиотеки моделирования процессов**, пожалуйста, обращайтесь к *Справочному руководству по Библиотеке моделирования процессов*.

Настройте блоки диаграммы

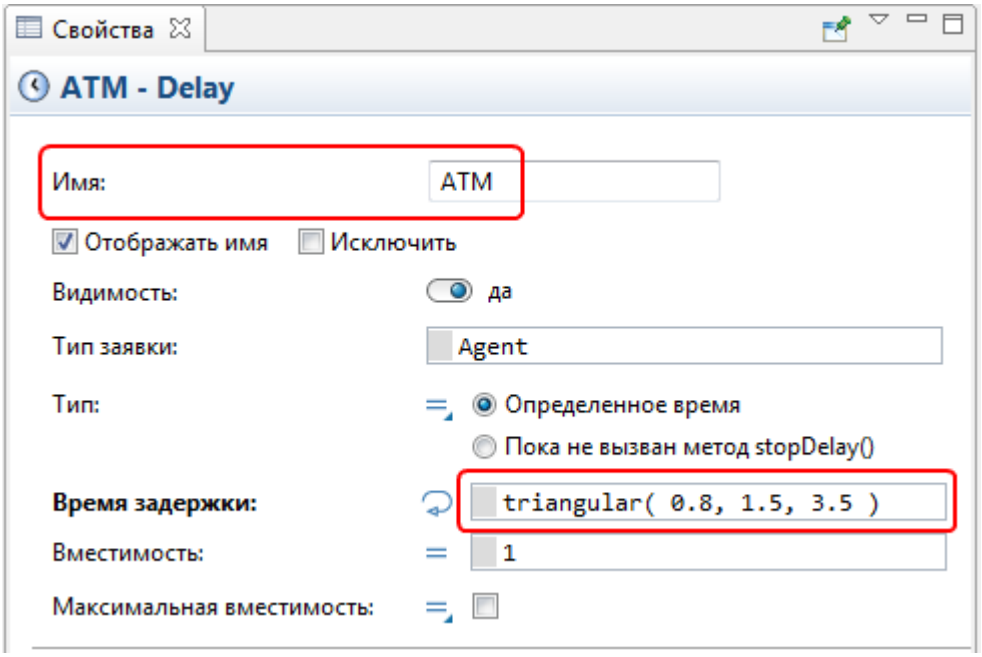
1. Чтобы изменить свойства элемента, выделите элемент в графическом редакторе или в панели **Проекты**, щелкнув по нему мышью. Свойства элемента откроются в панели **Свойства.**
2. Выделите блок *source*. В панели **Свойства** укажите, как часто должны прибывать клиенты. Введите *0.67* в поле **Интенсивность прибытия.**



3. Измените свойства блока *queue*. Введите в поле **Вместимость** *15*. В очереди будут находиться не более 15 человек.



4. Измените свойства блока *delay*. Назовите объект *ATM*. Задайте время обслуживания в поле **Время задержки**, распределенное по треугольному закону со средним значением, равным *1.5*, минимальным - равным *0.8* и максимальным - *3.5* минутам.



Функция triangular() является стандартной функцией генератора случайных чисел AnyLogic. AnyLogic предоставляет функции и других случайных распределений, таких как нормальное, равномерное, треугольное, и т.д.

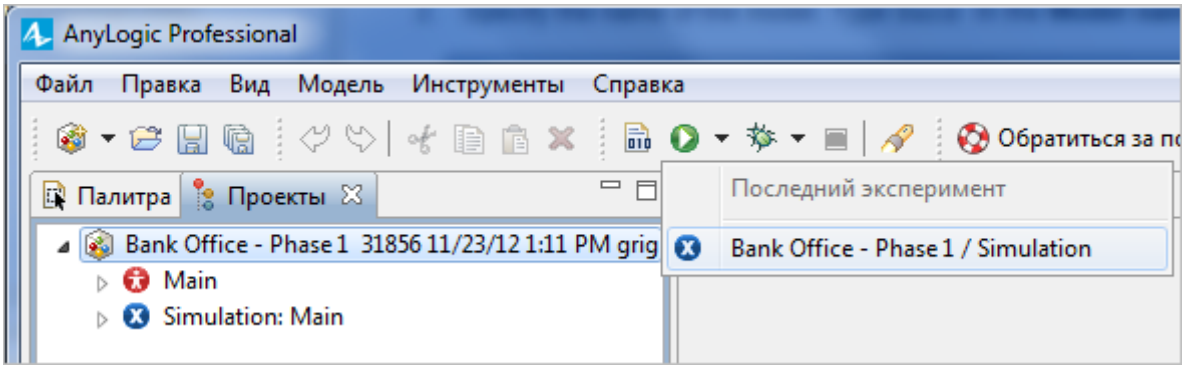
Запуск модели

Мы закончили моделирование простейшей системы очереди и готовы запустить созданную модель. Сначала постройте Вашу модель с помощью кнопки панели инструментов **Построить модель** (при этом в рабочей области AnyLogic должен быть выбран какой-то элемент именно этой модели). Если в модели есть какие-нибудь ошибки, то построение не будет завершено, и в панель **Ошибки** будет выведена информация об ошибках, обнаруженных в модели. Двойным щелчком мыши по ошибке в этом списке Вы можете перейти к предполагаемому месту ошибки, чтобы исправить ее.

После того, как Вы исправите все ошибки и успешно построите Вашу модель, Вы можете ее запустить. Запуская модель, вы автоматически обновляете ее.

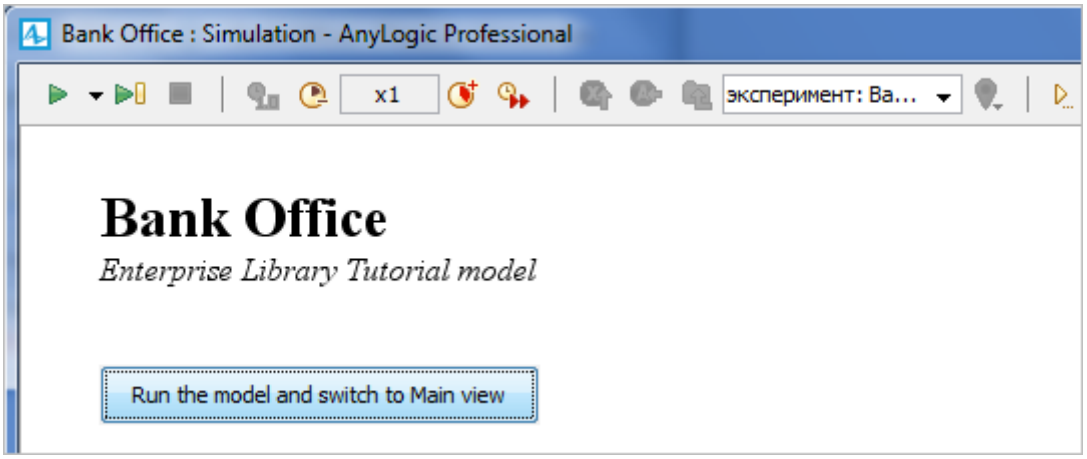
Запустите модель

1. Щелкните мышью по кнопке панели инструментов **Запустить** и выберите из открывшегося списка эксперимент, который Вы хотите запустить. Эксперимент этой модели будет называться Bank/Simulation.

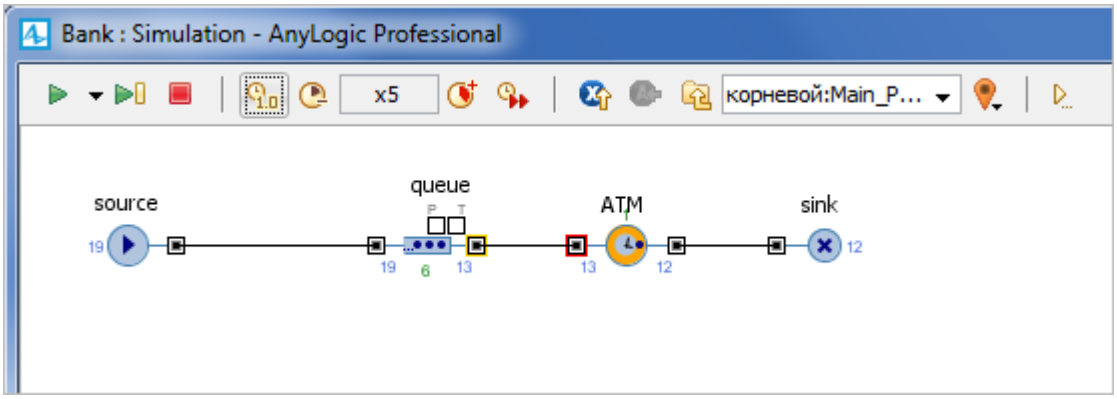


На момент запуска этого конкретного эксперимента наша модель - единственная открытая модель в рабочем пространстве. В дальнейшем будет запускаться тот эксперимент, который запускался Вами в последний раз. Чтобы выбрать какой-то другой эксперимент, Вам будет нужно щелкнуть правой кнопкой мыши по этому эксперименту в панели **Проекты** и выбрать **Запустить** из контекстного меню.

Запустив модель, Вы увидите окно презентации этой модели. В нем будет отображена презентация запущенного эксперимента.



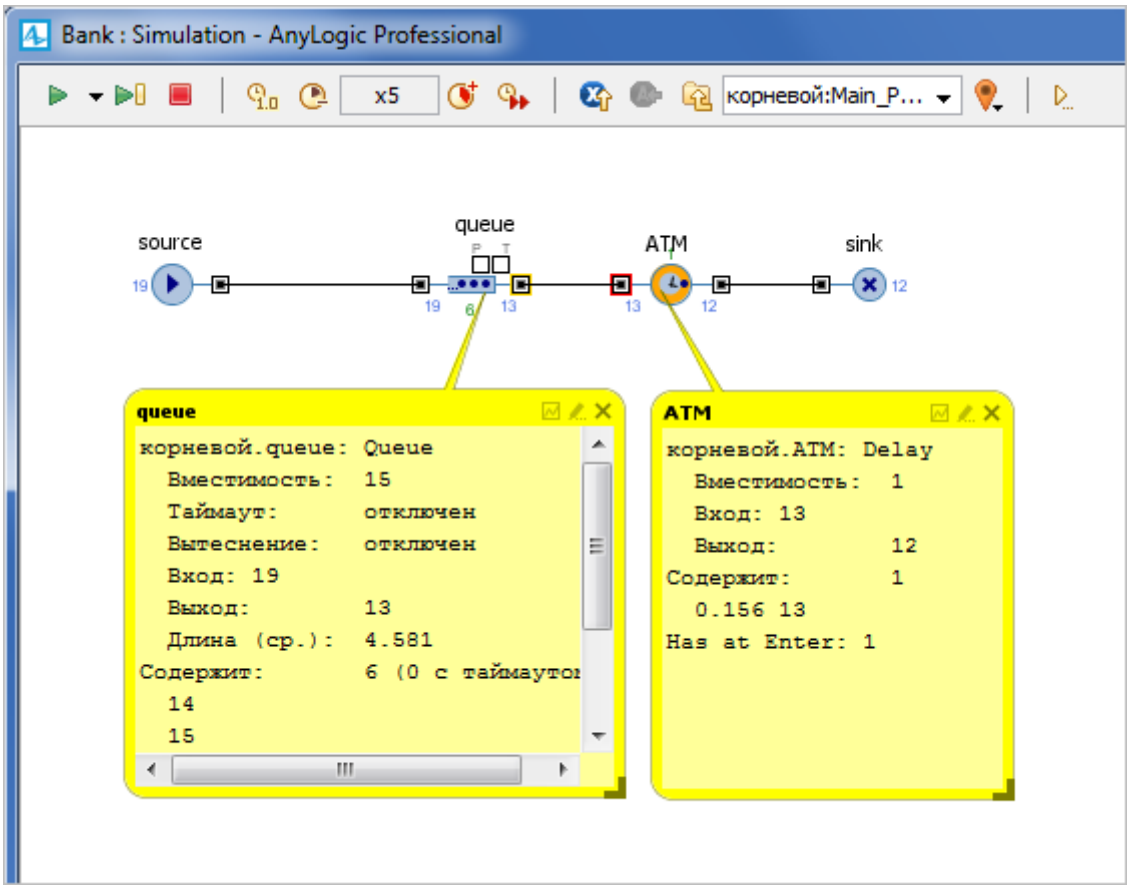
Щелкните по кнопке **Запустить модель**. Тем самым, Вы запустите модель и перейдете к презентации агента верхнего уровня запущенного эксперимента. Для каждой модели, созданной с помощью объектов **Библиотеки моделирования процессов**, автоматически создается блок-схема с наглядной визуализацией процесса, с помощью которой Вы можете изучить текущее состояние модели, например, длину очереди, количество обслуженных человек и так далее.



Вы можете изменить скорость выполнения модели с помощью кнопок панели инструментов **Замедлить** и **Ускорить**.

Вы можете следить за состоянием любого блока диаграммы процесса во время выполнения модели с помощью окна инспекта этого объекта. Чтобы открыть окно инспекта, щелкните мышью по значку блока. В окне инспекта будет отображена базовая инормация по выделенному блоку: например, для блока **Queue** будет отображена вместимость очереди, количество заявок, прошедшее через каждый порт объекта, и т.д.

Строка *Содержит* отображает количество заявок, находящихся в данный момент на объекте вместе с ID этих заявок.



3.2. Создание анимации модели

Хотя мы и могли анализировать работу запущенной нами только что модели с помощью диаграммы процесса, но куда удобнее было бы иметь более наглядную анимацию моделируемого нами с помощью анимации. В этом примере мы хотим создать визуализированный план банковского отделения.

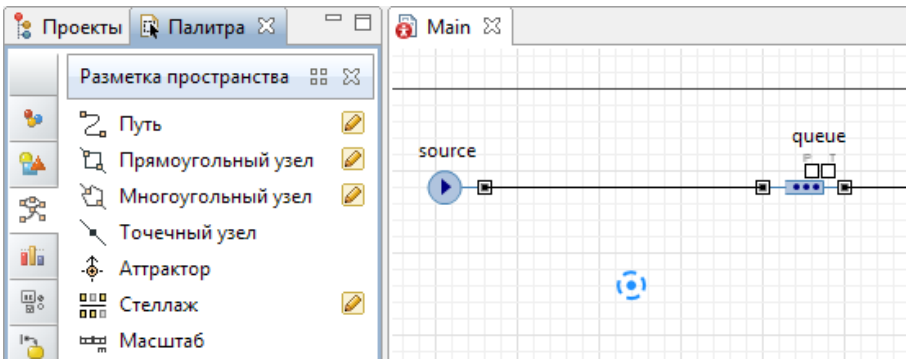
Поскольку в нашем случае нас не интересует конкретное расположение объектов в пространстве, то мы можем просто добавить чисто схематическую анимацию интересующих нас объектов - в нашем случае мы хотим видеть на анимации банкомат и ведущую к нему очередь клиентов.

Анимация модели рисуется в той же диаграмме (в графическом редакторе), в которой задается и диаграмма моделируемого процесса.

Добавление фигур разметки пространства

Задайте фигуру анимации банкомата

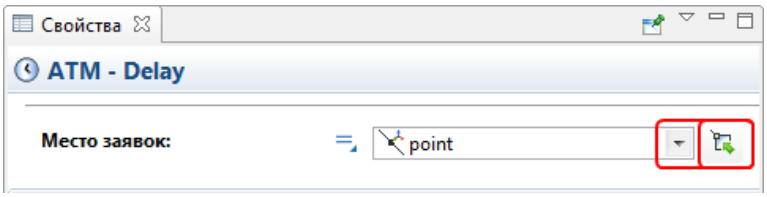
1. Нарисуем точечный узел, обозначающий банкомат. Вначале откройте палитру **Разметка пространства** панели **Палитра**.
2. Перетащите элемент **Точечный узел** из палитры **Разметка пространства** в графический редактор и поместите его под блок-схемой процесса.



3. Выделите щелчком точечный узел в графическом редакторе, чтобы открыть для него панель **Свойства**. Мы с Вами хотим, чтобы во время моделирования менялся цвет нашей фигуры, поэтому введите выражение, которое будет постоянно вычисляться заново при выполнении модели, в поле **Цвет**:
   ATM.size()> 0 ? red : green

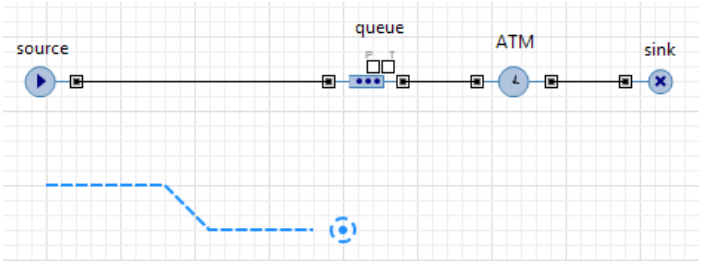
Здесь ATM – это имя нашего объекта *Delay*. Функция size() возвращает число человек, обслуживаемых в данный момент времени. Если банкомат занят, то цвет кружка будет красным, в противном случае - зеленым.

4. Выделите щелчком блок *delay*, названный нами ATM в диаграмме процесса, чтобы открыть его свойства.
5. Выберите точечный узел *point*, который мы только нарисовали в параметре **Место заявок**. Вы можете выбрать его из выпадающего списка подходящих объектов, щелкнув стрелку "вниз", или выбрать фигуру из графического редактора, предварительно щелкнув кнопку справа от поля параметра (в таком случае все неподходящие объекты в графическом редакторе будут обесцвечены).

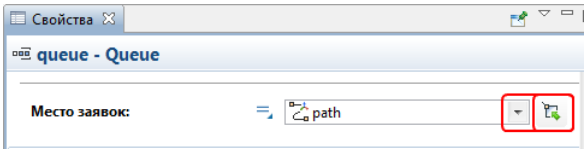


Задайте фигуру анимации очереди к банкомату

1. Нарисуем путь, обозначающий очередь к банкомату. Вначале откройте палитру **Разметка пространства** панели **Палитра**.
2. Двойным щелчком выделите элемент **Путь** палитры **Разметка пространства**, чтобы перейти в *режим рисования*.
3. Теперь Вы можете рисовать путь точка за точкой, последовательно щелкая мышью в тех точках диаграммы, куда Вы хотите поместить вершины ломаной. Чтобы завершить рисование, добавьте последнюю точку ломаной двойным щелчком мыши.

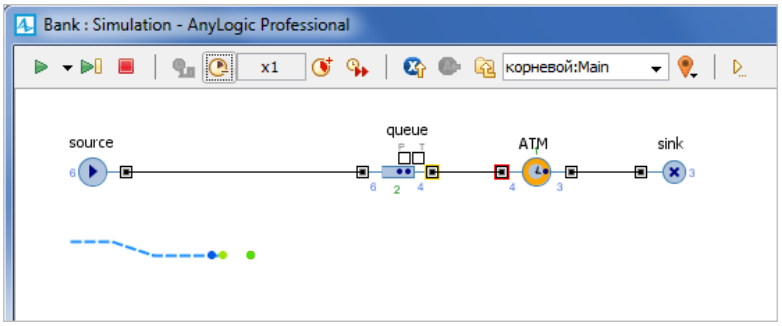


4. Выделите щелчком блок *queue* в диаграмме процесса, чтобы открыть для него панель **Свойства**.
5. Выберите путь *path*, который мы только нарисовали в параметре **Место заявок**. Вы можете выбрать его из выпадающего списка подходящих объектов, щелкнув стрелку "вниз", или выбрать фигуру из графического редактора, предварительно щелкнув кнопку справа от поля параметра (в таком случае все неподходящие объекты в графическом редакторе будут обесцвечены).



Теперь Вы можете запустить модель и изучить ее поведение. Для ускорения работы модели, переключитесь в режим виртуального времени, щелкнув мышью по кнопке панели инструментов **Реальное/виртуальное время**. В режиме виртуального времени модель будет выполняться максимально быстро, без привязки модельного времени к реальному.

Запустите модель. Вы увидите, что у нашей модели теперь есть простейшая анимация - банкомат и ведущую к нему очередь клиентов. Цвет фигуры банкомата будет меняться в зависимости от того, обслуживается ли клиент в данный момент времени.



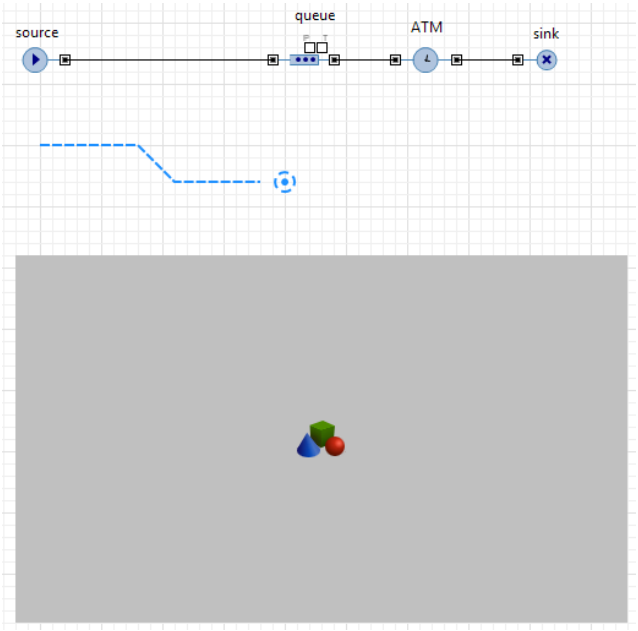
Добавление 3D анимации

Первым делом нам будет нужно добавить на диаграмму активного объекта 3D Окно.

3D Окно используется для задания на диаграмме агента области, в которой во время запуска модели будет отображаться трехмерная анимация этой модели.

Добавьте 3D окно

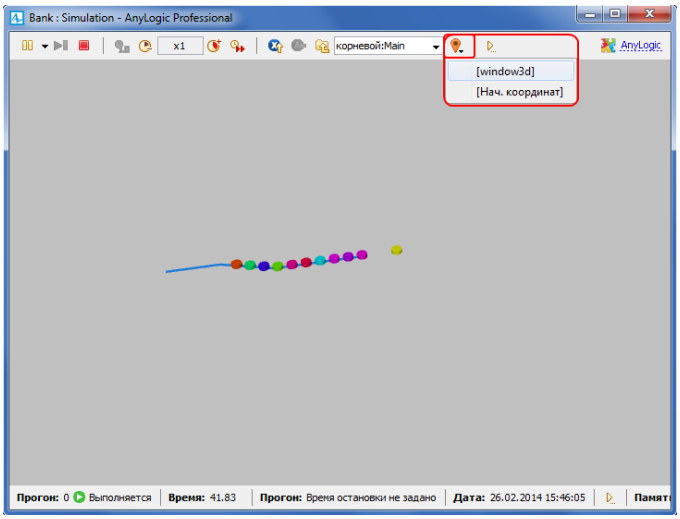
1. Перетащите элемент **3D Окно** из секции **3D** палитры **Презентация** в графический редактор.
2. Вы увидите в графическом редакторе закрашенную серым область. Поместите ее туда, где Вы хотите видеть 3D анимацию во время прогона модели:



Запустите модель и опробуйте навигацию по сцене трехмерной анимации

Мы создали простейшую трехмерную анимацию и готовы к тому, чтобы запустить модель и посмотреть на результат нашей работы.

1. Щелкните кнопку панели инструментов **Показать область...** и выберете **[window3D]**.



2. Попробуйте "подвигаться" по трехмерной сцене с помощью описанных ниже команд навигации:

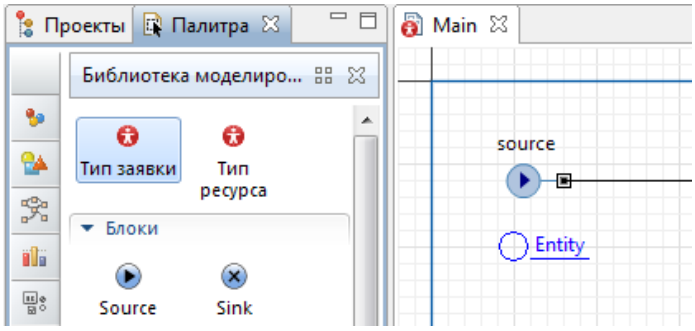
Действие	Манипуляции мышью
Переместить сцену	1. Нажмите левую кнопку мыши в области 3D окна и держите ее нажатой. 2. Передвиньте мышь в направлении перемещения.
Повернуть сцену	1. Нажмите клавишу **Alt** и держите ее нажатой. 2. Нажмите левую кнопку мыши в области 3D окна и держите ее нажатой. 3. Передвиньте мышь в направлении вращения.
Приблизить/отдалить сцену	1. Покрутите колесо мыши от/на себя в области 3D окна.

Добавление 3D объектов

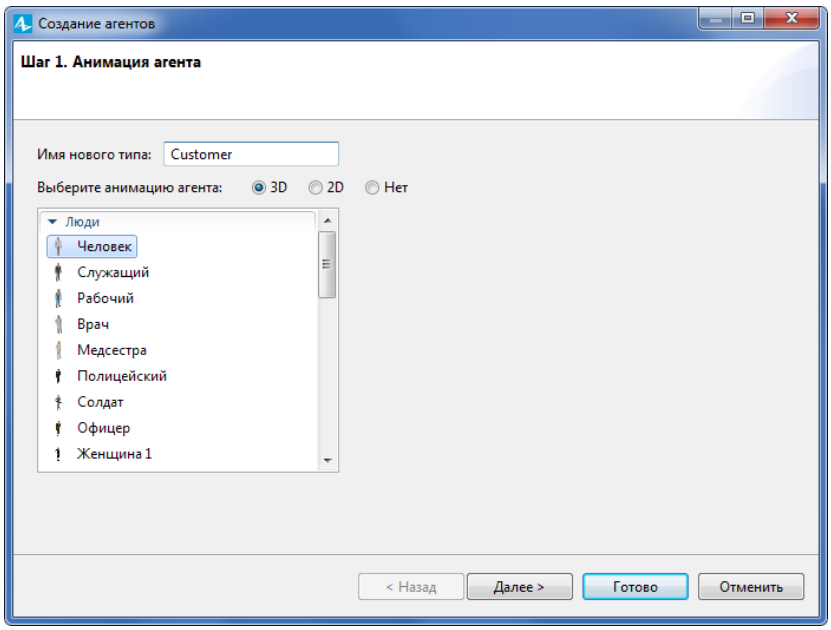
Теперь мы хотим задать фигуру клиента банка. По умолчанию клиенты в нашей модели обозначались цветными точками и отображались цветными цилиндрами в 3D анимации. Если мы хотим задать нестандартный тип клиента и выбрать для него красивую фигуру анимации, нам нужно создать новый тип заявки.

Создайте новый тип заявки

1. Откройте **Библиотеку моделирования процессов** в панели **Палитра**.
2. Перетащите элемент **Тип заявки** в графический редактор.



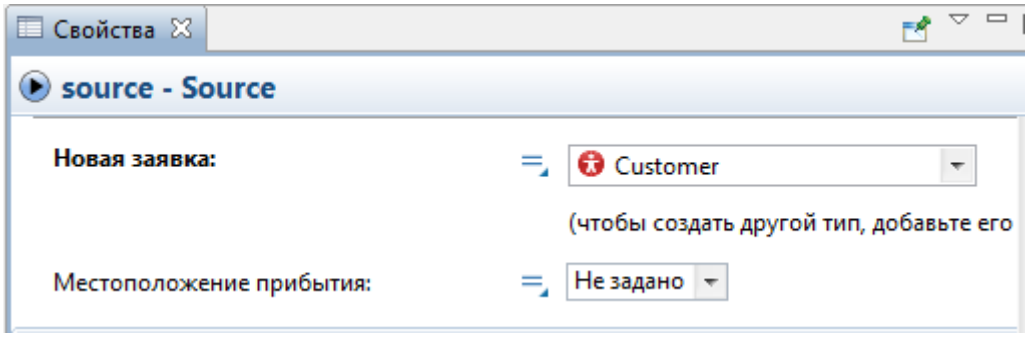
3. Откроется диалоговое окно Мастера создания агентов на шаге **Анимация агента**. Введите *Customer* в поле **Имя нового типа**, выберите опцию **3D** для типа анимации и фигуру анимации *Человек* из списка 3D фигур.



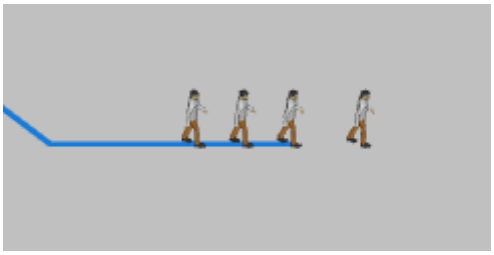
4. Щелкните **Готово**. Новая диаграмма Customer автоматически откроется. Вы можете найти 3D фигуру *Человек* в начале координат.

Настройте использование нового типа заявок в блок-схеме

1. На диаграмме Main, выделите блок *source* в графическом редакторе.
2. Выберите тип заявок Customer в выпадающем списке параметра **Новая заявка**.



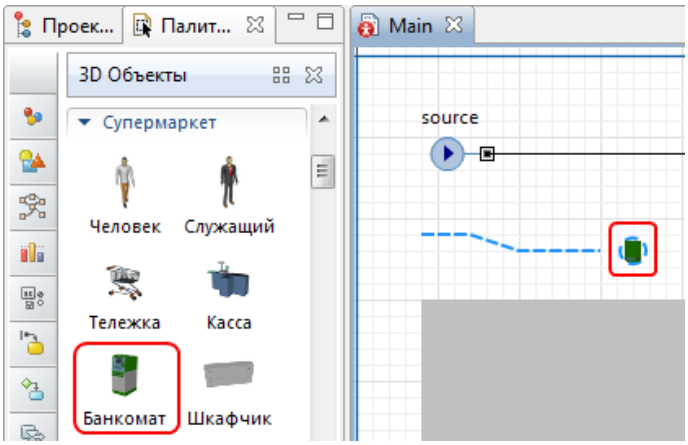
3. Запустите модель, чтобы увидеть анимацию клиентов в очереди.



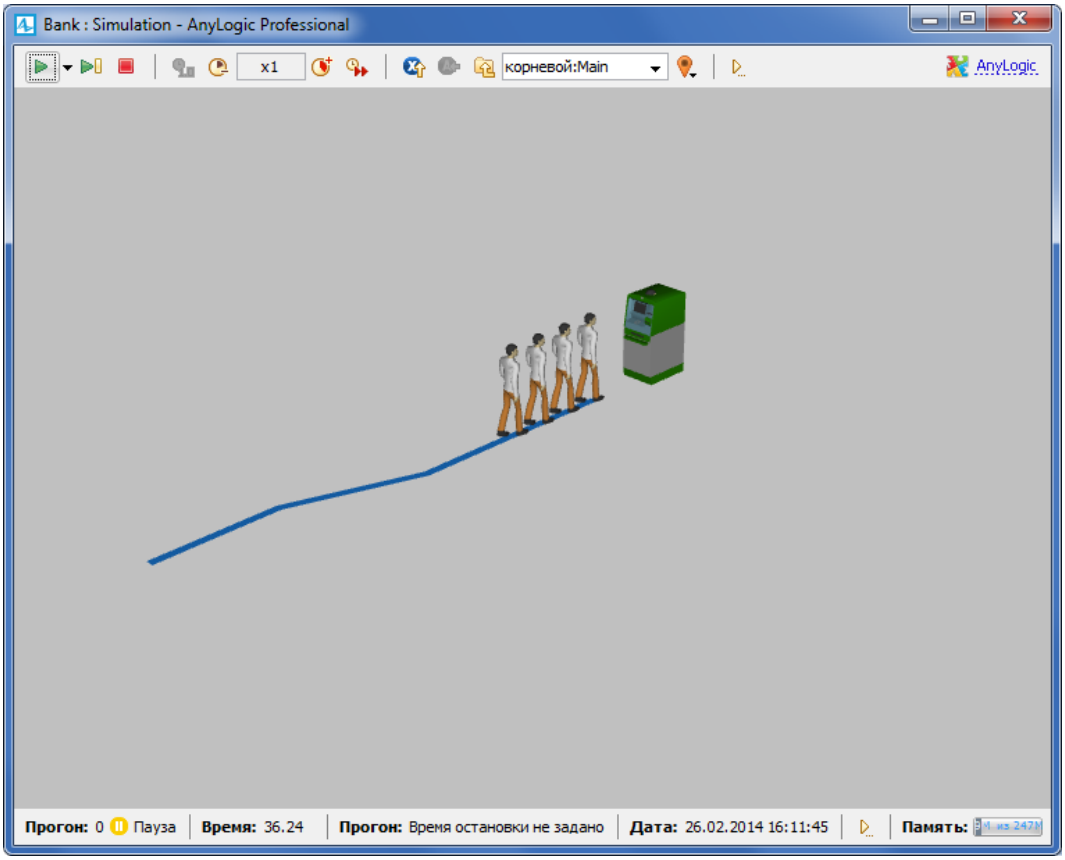
Добавьте объект банкомата

1. Откройте палитру **3D Объекты** в панели **Палитра**.

2. Перетащите 3D фигуру **Банкомат** из секции палитры **Супермаркет** в графический редактор и поместите ее на точечный узел.



3. Если Вы сейчас запустите модель и проверите 3D анимацию в режите просмотра **window3D**, Вы заметите, что банкомат стоит не той стороной по направлению к очереди клиентов, и нам необходимо развернуть его в правильную сторону.
4. Выделите 3D объект банкомата *atm* в графическом редакторе и откройте секцию свойств **Расположение**.
5. Выберите из выпадающего списка параметра **Поворот Z** *0* градусов.
6. Запустите модель, чтобы убедиться, что фигура банкомата стоит "лицом" к клиентам.



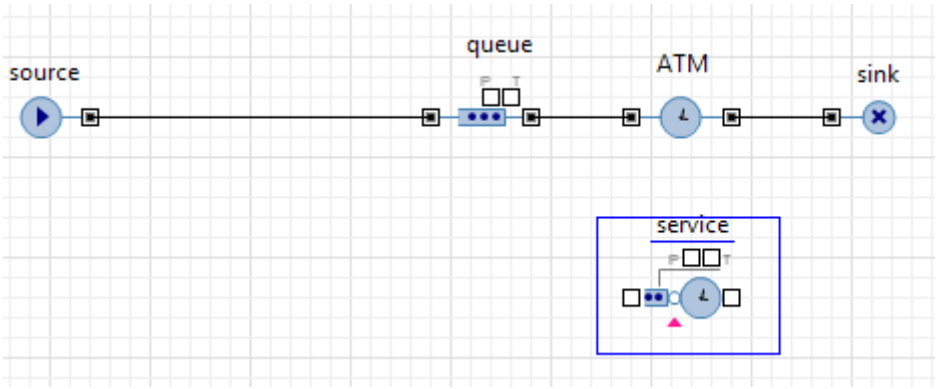
3.3. Добавление клерков

Теперь мы усложним нашу модель, добавив в нее служащих – банковских кассиров. Мы могли бы промоделировать кассиров, как и банкомат, с помощью объектов **Delay**. Но куда более удобным представляется моделирование кассиров с помощью ресурсов. Ресурс – это специальный объект **Библиотеки моделирования процессов**, который может потребоваться заявке для выполнения какой-то задачи. В каждый момент времени ресурс может быть занят только одной заявкой. В нашем примере посетителям банковского отделения (заявкам) необходимо получить помощь у банковских служащих (ресурсов).

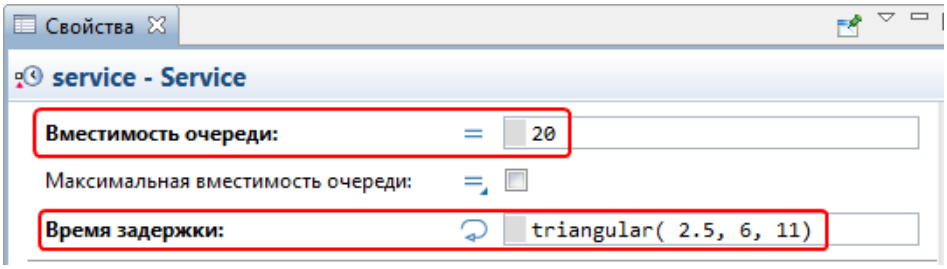
Изменения в диаграмме процесса

Добавьте обслуживание

1. Откройте **Библиотеку моделирования процессов** в панели **Палитра** и перетащите на диаграмму процесса Main блок **Service**.
   Объект **Service** захватывает для заявки заданное количество ресурсов, задерживает заявку, а затем освобождает захваченные ею ресурсы.

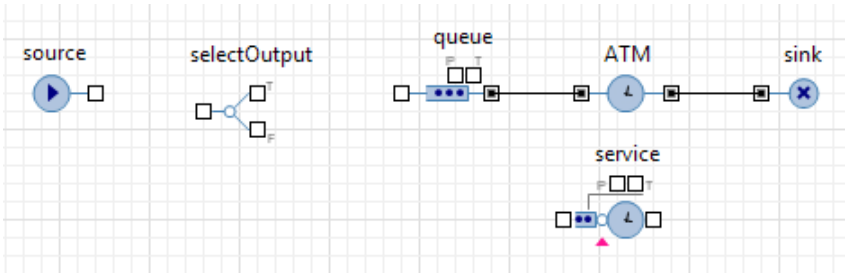


2. Перейдите в панель **Свойства** блока *service*.
3. Измените параметры объекта следующим образом:
   - Ко всем кассирам будет вести одна общая очередь. Задайте максимальное количество человек в этой очереди в поле **Вместимость очереди**: *20*.
   - Мы полагаем, что время обслуживания имеет треугольное распределение с минимальным значением равным *2.5*, средним - *6*, и максимальным - *11* минутам. Введите в поле **Время задержки**: triangular(2.5, 6, 11)



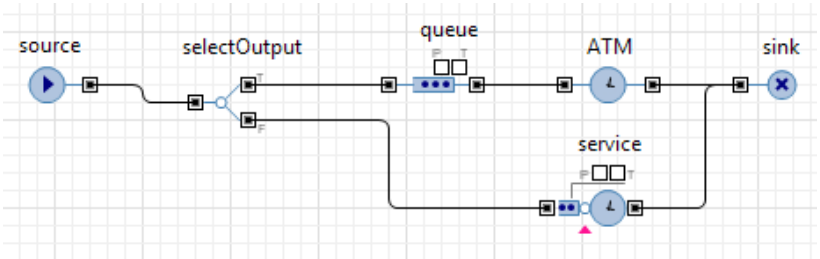
Смоделируйте выбор клиентов

1. Удалите соединитель между блоками *source* и *queue* в диаграмме процесса.
2. Откройте **Библиотеку моделирования процессов** в панели **Палитра** и перетащите на диаграмму процесса Main блок **SelectOutput** в образовавшееся свободное место.
   SelectOutput является блоком принятия решения. В зависимости от заданного Вами условия, заявка, поступившая в объект, будет поступать на один из двух выходных портов объекта.



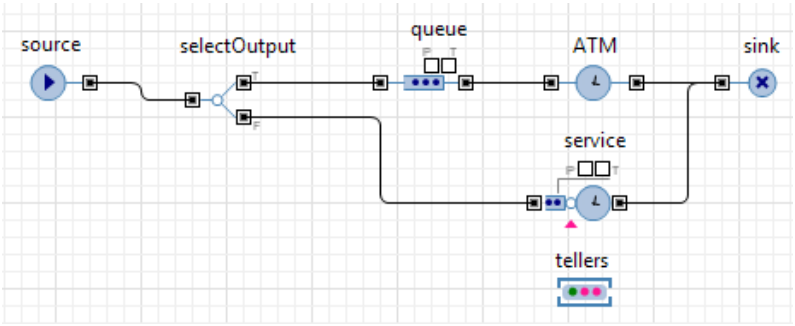
3. Выделите блок *selectOutput* в диаграмме процесса. В панели **Свойства** этого блока выберите опцию *При выполнении условия* в параметре **Выход True выбирается**. Убедитесь, что в поле **Условие** стоит выражение randomTrue(0.5).
   В этом случае к кассирам и банкомату будет приходить примерно равное количество клиентов.

4. Соедините блоки *selectOutput* и *service* с дургими блоками так, как показано на рисунке ниже:

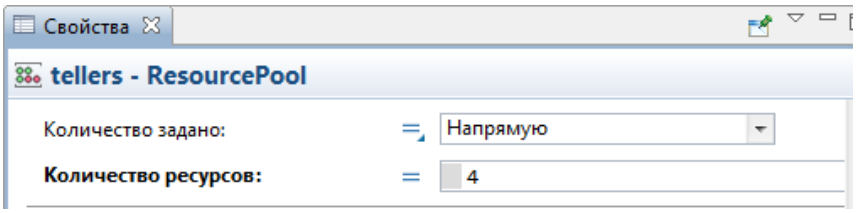


Добавьте ресурсы для сервиса

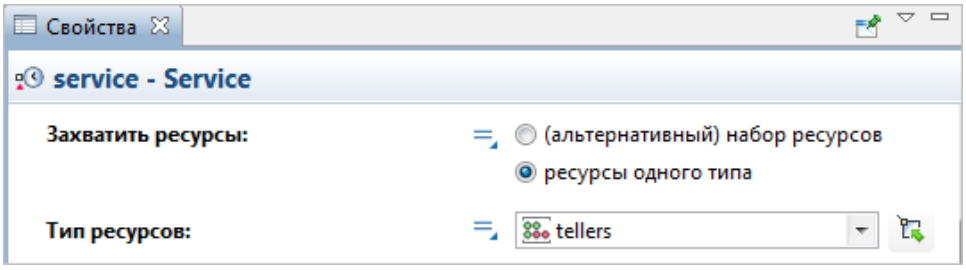
1. Откройте **Библиотеку моделирования процессов** в панели **Палитра** и перетащите блок **ResourcePool** на диаграмму процесса Main. Объект **ResourcePool** задает ресурсы определенного типа (в нашей модели это будут банковские клерки).
2. Поместите его, например, под блоком *service* и перейдите в панель **Свойства**.
3. Назовите объект *tellers*.



4. Задайте число кассиров в поле **Кол-во ресурсов**: *4*.



5. Блок **ResourcePool** указывается в объектах, использующих ресурсы, в нашем случае это блок **Service**. Поэтому нам необходимо изменить свойства блока *service* диаграмму процесса.
6. Выделите блок *service* и перейдите в панель **Свойства**. Выберите опцию *Ресурсы одного типа* в параметре **Захватить ресурсы**. Затем укажите блок *tellers*, который мы добавили на диаграмму, в параметре **Блок ResourcePool**. Вы можете выбрать его из выпадающего списка подходящих объектов, щелкнув стрелку "вниз", или выбрать фигуру из графического редактора, предварительно щелкнув кнопку справа от поля параметра (в таком случае все неподходящие объекты в графическом редакторе будут обесцвечены).



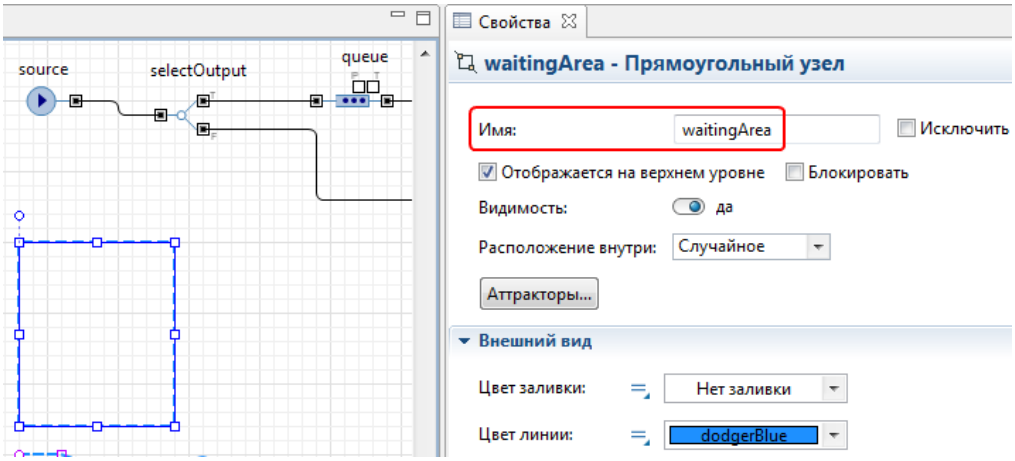
7. Поскольку наша модель изменилась, мы должны изменить и ее анимацию.

Добавление фигур разметки пространства

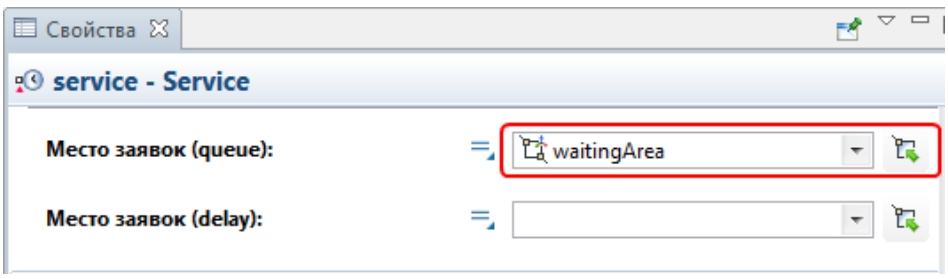
Теперь давайте нарисуем область для ожидания и место обслуживания клиентов кассирами.

Задайте фигуру разметки для электронной очереди

1. В этот раз мы будем рисовать место ожидания клиентами, используя прямоугольный узел. Вначале откройте палитру **Разметка пространства** панели **Палитра**.
2. Двойным щелчком выделите элемент **Прямоугольный узел** палитры **Разметка пространства**, чтобы перейти в *режим рисования*.
3. Щелкните мышью в графическом редакторе, чтобы задать вершину верхнего левого угла, затем тащите прямоугольник, не отпуская кнопки мыши. Отпустите, когда прямоугольный узел имеет нужную форму. Вы можете редактировать фигуру и после того, как ее рисование завершено
4. Назовите эту область *waitingArea*.

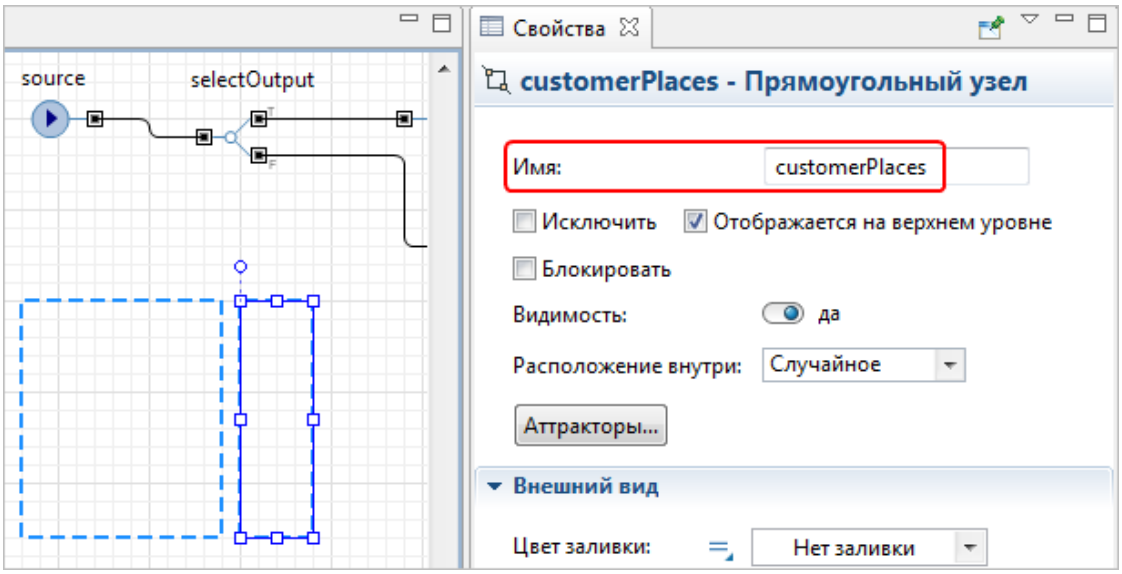


5. Выделите щелчком блок *service* block в диаграмме процесса и перейдите в его свойства.
6. Выберите только что нарисованный нами узел *waitingArea* в параметре **Место заявок (queue)**.

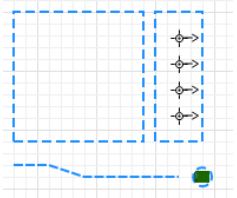


Задайте фигуру разметки места обслуживания клиентов

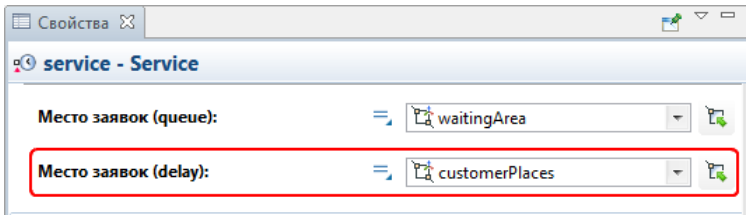
1. Клиентам банка требуется место, на котором они могли бы находиться во время обслуживания у кассиров. Мы нарисуем такую область, используя прямоугольный узел.
   2. Вначале откройте палитру **Разметка пространства** панели **Палитра**.
   3. Двойным щелчком выделите элемент **Прямоугольный узел** палитры **Разметка пространства**, чтобы перейти в *режим рисования*.
   4. Щелкните мышью в графическом редакторе, чтобы задать вершину верхнего левого угла, затем тащите прямоугольник, не отпуская кнопки мыши. Отпустите, когда прямоугольный узел имеет нужную форму. Вы можете редактировать фигуру и после того, как ее рисование завершено.
   5. Назовите эту область *customerPlaces*.



   6. Мы будем использовать аттракторы, чтобы задать местоположение тех клиентов, которые будут обслуживаться у клерков. Выделите узел *customerPlaces* в графическом редакторе и щелкните кнопку **Аттракторы...** в свойствах узла. В открывшемся окне **Аттракторы** укажите число аттракторов *4* в режиме создания **Количество аттракторов**, затем щелкните **OK**. Вы увидите, что четыре аттрактора появились в узле *customerPlaces* на равном расстоянии друг от друга.

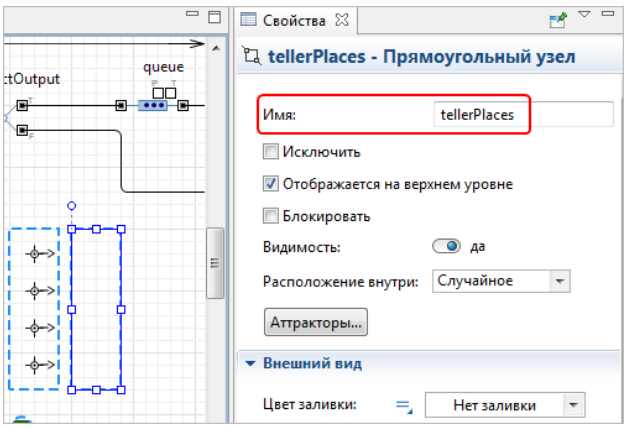


7. Теперь нам необходимо сослаться на эту фигуру в диаграмме процесса. Щелкните блок *service* и перейдите в панель **Свойства** этого блока.
8. Выберите нарисованный нами узел *customerPlaces* в параметре **Место заявок (delay)**.



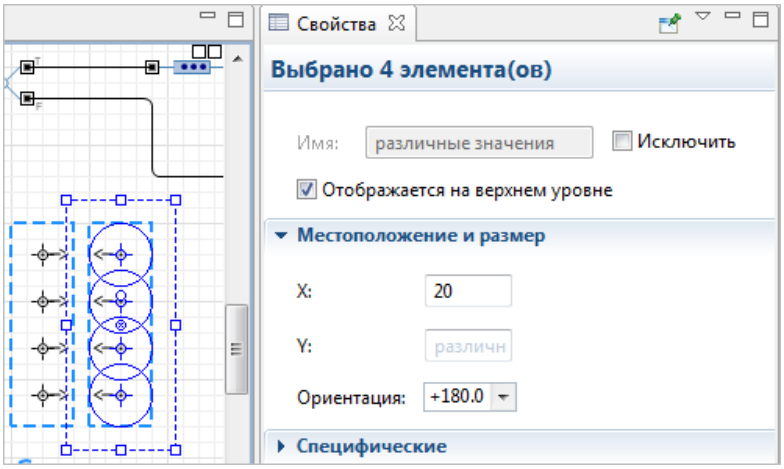
Задайте фигуру разметки для кассиров

1. Клиентам банка требуется место, на котором они могли бы находиться во время обслуживания у кассиров. мы нарисуем такую область, используя прямоугольный узел.
   2. Вначале откройте палитру **Разметка пространства** панели **Палитра**.
   3. Двойным щелчком выделите элемент **Прямоугольный узел** палитры **Разметка пространства**, чтобы перейти в*режим рисования*.
   4. Щелкните мышью в графическом редакторе, чтобы задать вершину верхнего левого угла, затем тащите прямоугольник, не отпуская кнопки мыши. Отпустите, когда прямоугольный узел имеет нужную форму. Вы можете редактировать фигуру и после того, как ее рисование завершено.
   5. Назовите эту область *tellerPlaces*.

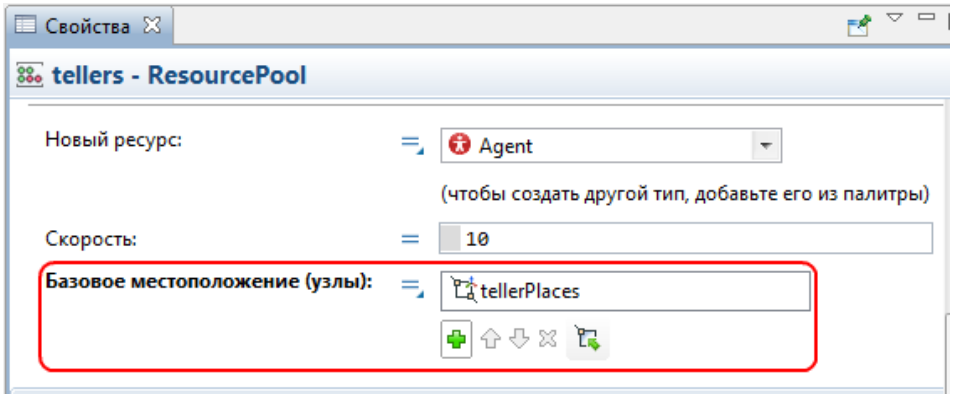


6. Мы будем использовать аттракторы, чтобы задать местоположение клерков. Выделите узел *tellerPlaces* в графическом редакторе и щелкните кнопку **Аттракторы...** в свойствах узла. В открывшемся окне **Аттракторы** укажите число аттракторов *4* в режиме создания **Количество аттракторов**, затем щелкните **OK**.

7. Вы увидите, что четыре аттрактора появились в узле *customerPlaces* на равном расстоянии друг от друга, но они направлены не в ту сторону. Выделите все аттракторы Shift+щелчком и выберите *+180.0* в параметре **Ориентация** секции свойств **Местоположение и размер**.



8. Щелкните объект *tellers* в диаграмме процесса и перейдите в его свойства.
9. Выберите нарисованный нами узел *tellerPlaces* в параметре **Базовое местоположение (узлы)**.



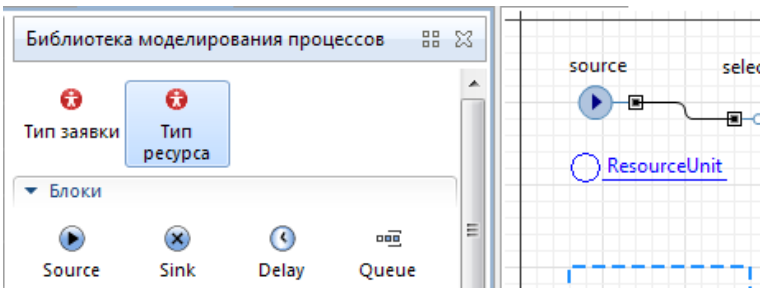
Вы можете запустить модель и наблюдать, как клиенты обслуживаются у банкоматов и проходят к кассирам.

Добавление 3D объектов

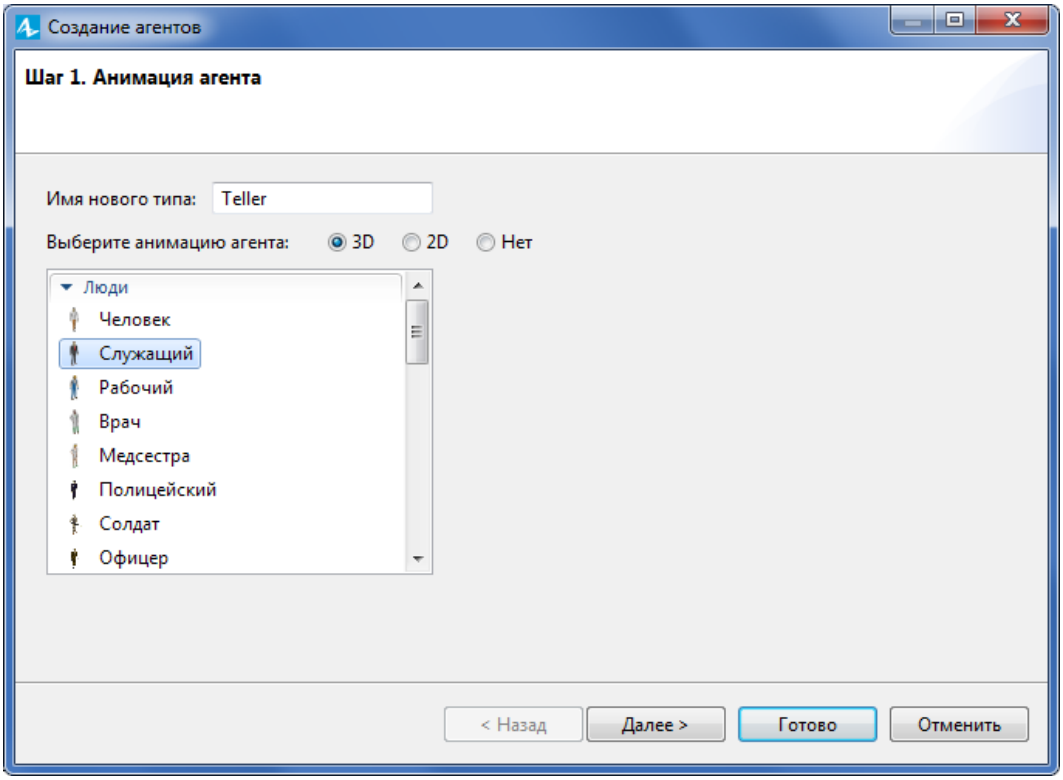
Давайте добавим 3D фигуры клерков в нашу модель. Мы создадим новый тип ресурсов для анимации клерков.

Создайте новый тип ресурсов

1. Откройте **Библиотеку моделирования процессов** в панели **Палитра**.
2. Перетащите элемент **Тип ресурса** в графический редактор.



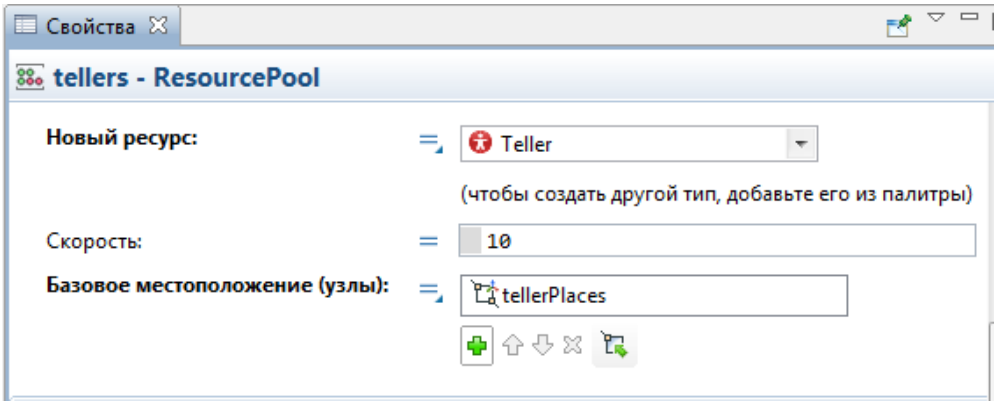
3. Откроется диалоговое окно Мастера создания агентов на шаге **Анимация агента**. Введите *Teller* в поле **Имя нового типа**, выберите опцию **3D** для типа анимации и фигуру анимации *Служащий* из списка 3D фигур.



4. Щелкните **Готово**. Новая диаграмма Teller автоматически откроется. Вы можете найти 3D фигуру *Служащий* в начале координат. Переключитесь обратно на диаграмму Main.

Настройте использование нового типа ресурсов в блок-схеме

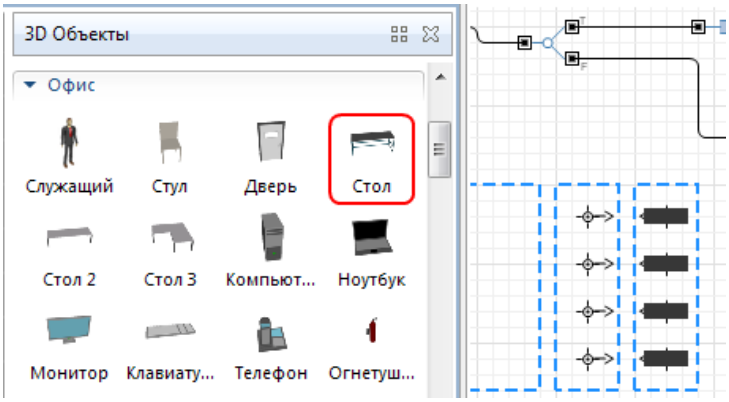
1. На диаграмме Main, выделите блок *tellers* в графическом редакторе.
2. Выберите тип ресурсов Teller в выпадающем списке параметра **Новый ресурс.**



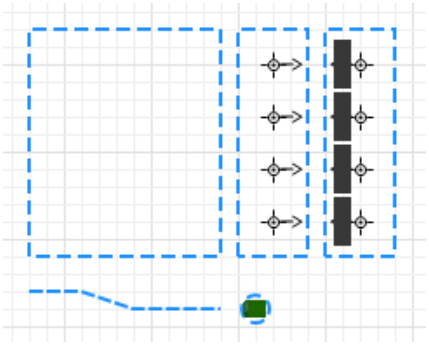
3. Запустите модель, чтобы увидеть получившуюся анимацию клерков.

Добавьте столы для клерков

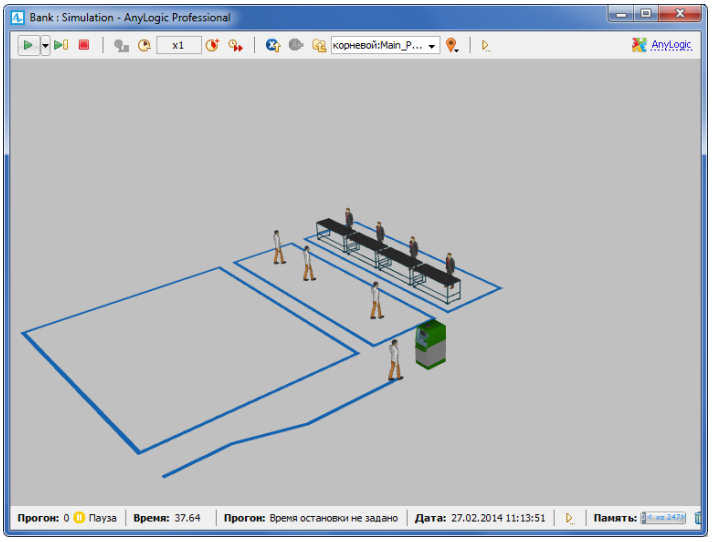
1. Откройте палитру **3D Объекты** в панели **Палитра**.
2. Перетащите четыре 3D фигуры **Стол** из секции палитры **Офис** в графический редактор и поместите их в узел *tellerPlaces.*
3. Расположите столы на аттракторах, так как аттракторы обозначают место, где стоят клерки



4. Вы заметите, что они стоят не той стороной к клеркам. Выделите все столы методом Shift-щелчок и перейдите в их свойства.
5. В секции **Расположение** измените параметр **Поворот Z**: *-90.0* градусов.
6. При необходимости, выровняйте расположение всех восьми аттракторов и столов.



Теперь Вы можете запустить модель и увидеть в 3D анимации, как некоторые клиенты идут к банкомату, а другие обслуживаются у столов клерков.



3.4. Добавление статистики модели

AnyLogic предоставляет пользователю удобные средства для сбора статистики по работе блоков диаграммы процесса. Объекты Библиотеки моделирования процессов самостоятельно производят сбор основной статистики. Все, что Вам нужно сделать - это включить сбор статистики для объекта.

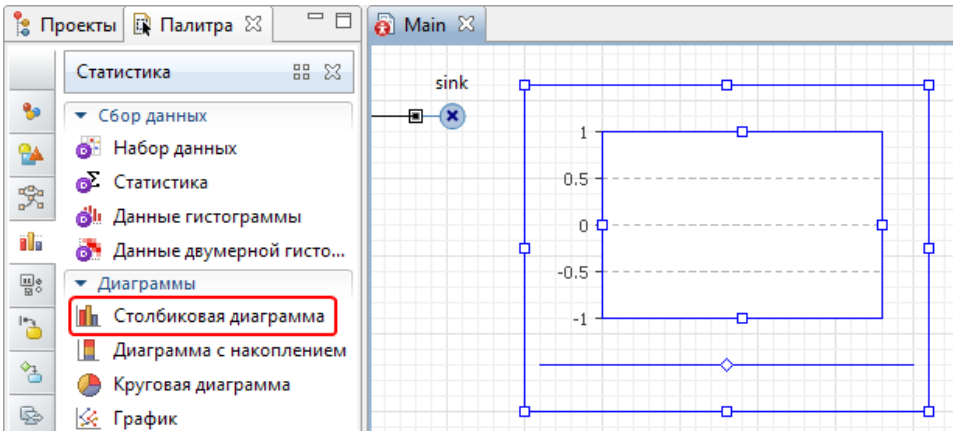
Мы можем, например, просмотреть интересующую нас статистику (скажем, статистику занятости банкомата и длины очереди) с помощью диаграмм.

Сбор статистики использования ресурсов

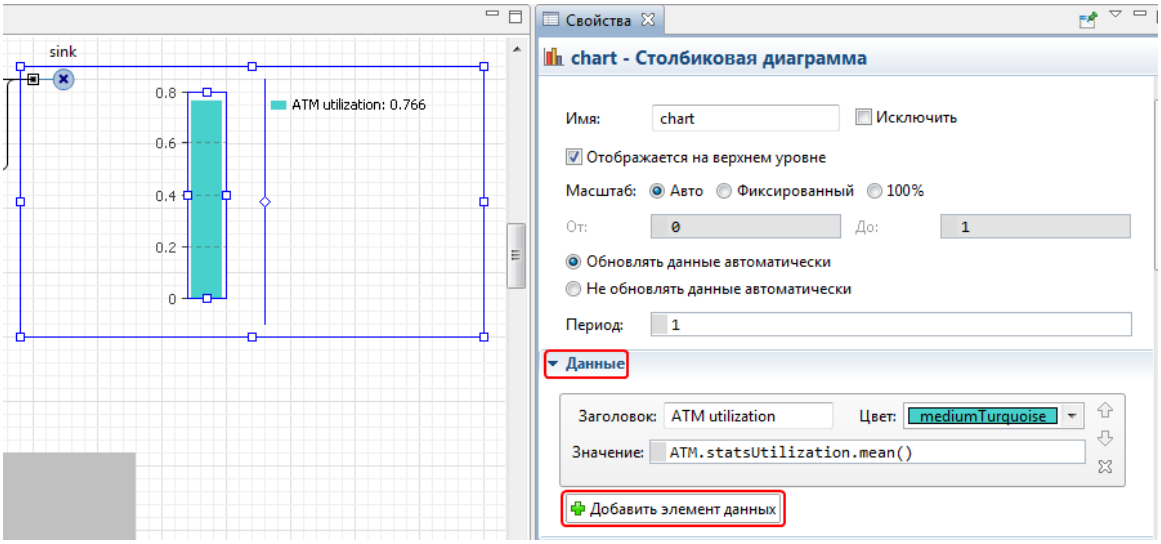
Добавьте диаграмму для отображения средней занятости банкомата

1. Откройте палитру **Статистика**. Эта палитра содержит элементы сбора данных и статистики, а также диаграммы для визуализации данных и результатов моделирования. Перетащите элемент **Столбиковая**

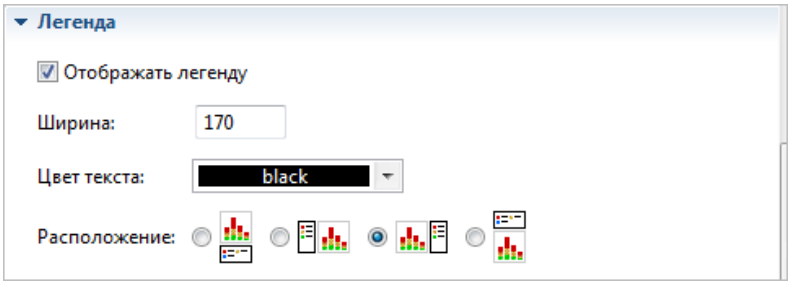
диаграмма из палитры **Статистика** на диаграмму и измените ее размер:



2. Перейдите в секцию **Данные** свойств столбиковой диаграммы. Щелкните кнопку **Добавить элемент данных**, чтобы задать данные для отображения в диаграмме.
3. Измените **Заголовок** на *ATM utilization*.
4. Введите ATM.statsUtilization.mean() в поле **Значение**. Здесь ATM - это имя нашего объекта **Dela**y. У каждого объекта **Delay** есть встроенный набор данных statsUtilization, занимающийся сбором статистики использования этого объекта. Функция mean() возвращает среднее из всех измеренных этим набором данных значений. Вы можете использовать и другие методы сбора статистики, такие, как min() или max().

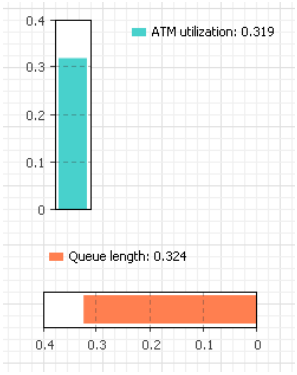


5. Перейдите в секцию **Легенда** панели **Свойства**. Измените расположение легенды относительно диаграммы (мы хотим, чтобы она отображалась справа).

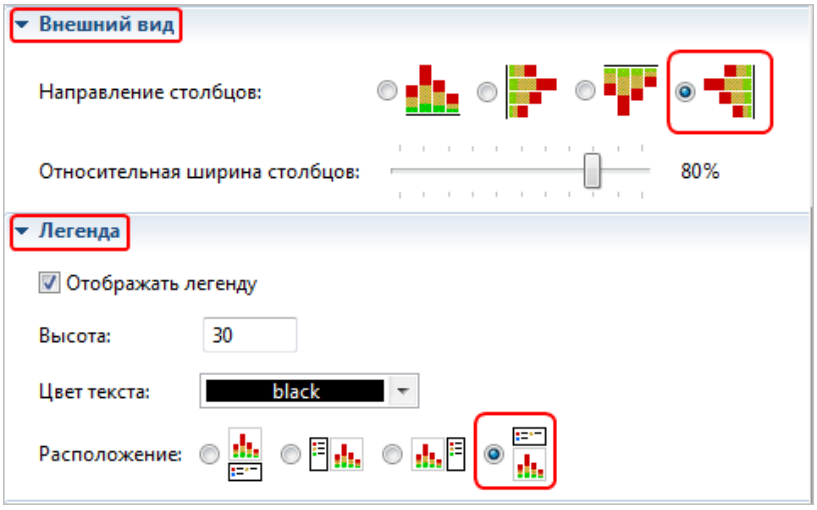


Добавьте диаграмму для отображения средней длины очереди

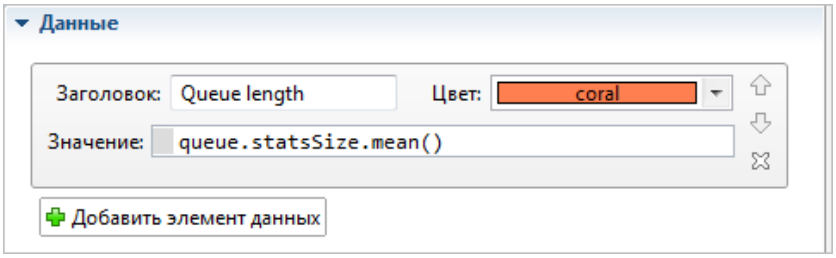
1. Аналогичным образом добавьте еще одну столбиковую диаграмму. Измените ее размер так, как показано на рисунке:



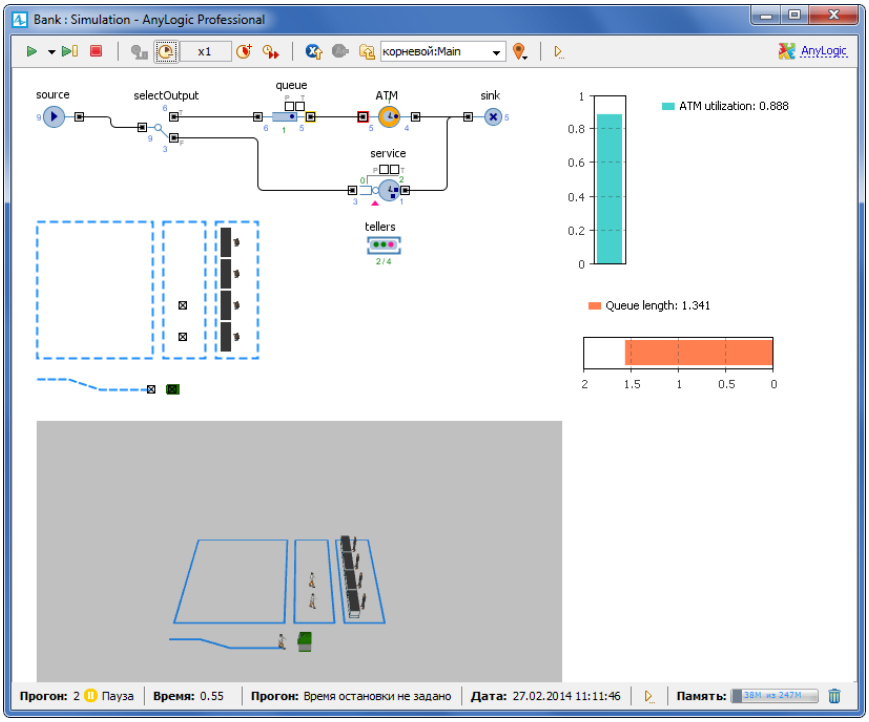
2. Перейдите в секцию **Внешний вид** панели **Свойства** и выберите последнюю опцию параметра **Направление столбцов**, чтобы столбцы столбиковой диаграммы росли влево. Также измените положение легенды в секции **Легенда** (как показано на рисунке ниже).



3. Добавьте элемент данных, который будет отображаться на диаграмме, в секции **Данные**. Задайте **Заголовок**: *Queue length* и задайте **Значение**: queue.statsSize.mean()
Здесь statsSize - это имя объекта типа "статистика" **StatisticsContinuous**, производящего сбор статистики размера очереди объекта **Queue**.



Запустите модель и пронаблюдайте за занятостью банкомата и средней длиной очереди с помощью только что созданных диаграмм.



Сбор статистики по времени обслуживания

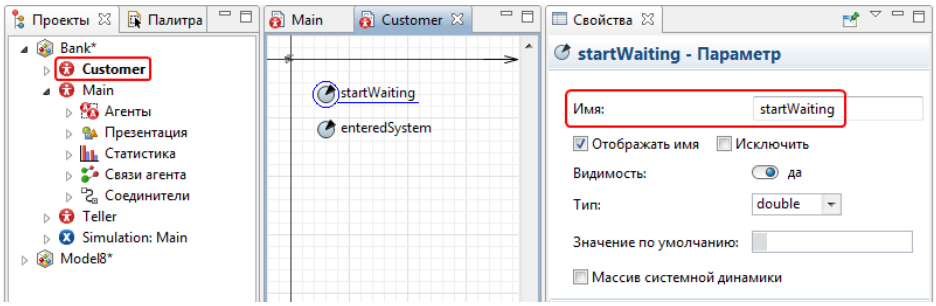
Мы хотим знать, сколько времени клиент проводит в банковском отделении и сколько времени он теряет, ожидая своей очереди. Мы соберем эту статистику с помощью специальных объектов сбора данных и отобразим собранную статистику распределения времен обслуживания клиентов с помощью гистограмм. Для этого мы будем использовать ранее созданный тип заявки Customer.

Вначале нам необходимо добавить два параметра в нашу модель.

Добавьте параметры

1. Переключитесь в панель **Проекты**. Дважды щелкните тип заявки Customer, чтобы открыть его диаграмму. Нам необходимо создать параметры на диаграмме агента Customer, так как мы хотим собирать статистику клиентов по времени их обслуживания.
2. Откройте палитру **Основная** в панели **Палитра**.

3. Перетащите два элемента **Параметр** на диаграмму Customer.
4. Назовите параметры *startWaiting* и *enteredSystem*, оставьте тип double, заданный по умолчанию

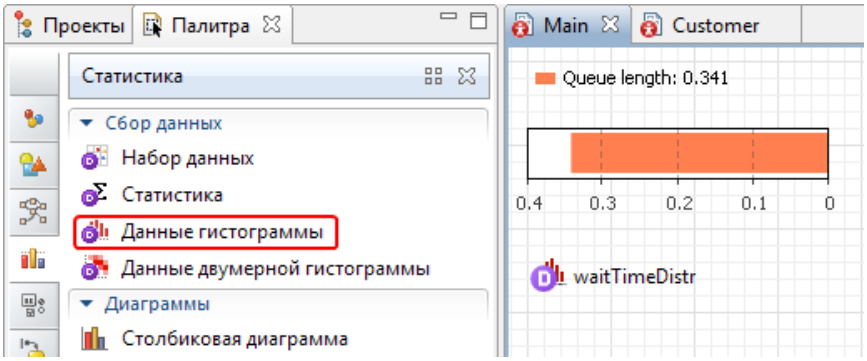


5. Мы продолжим разрабатывать нашу модель на диаграмме Main.

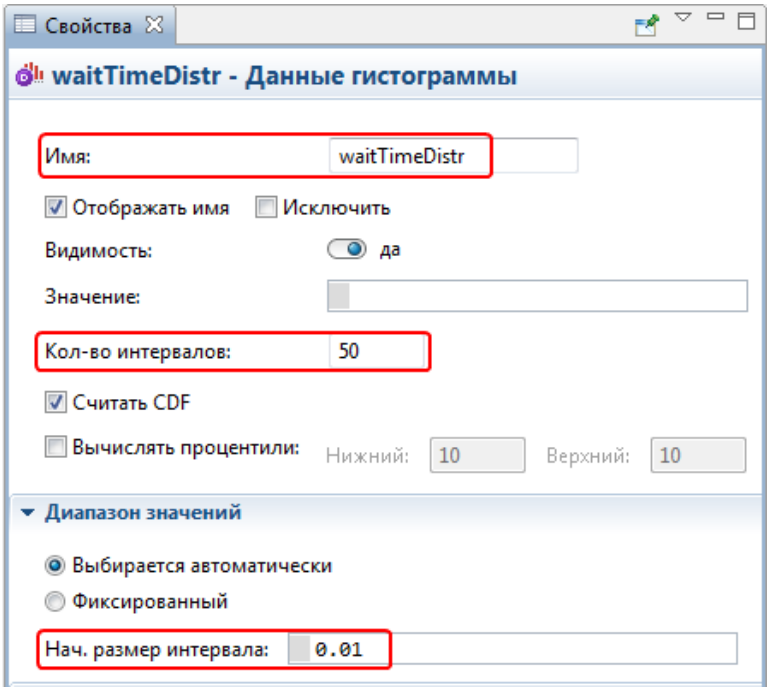
Добавьте элементы сбора статистики по времени ожидания клиентов и времени пребывания клиентов в системе. Эти элементы будут запоминать соответствующие значения времен для каждого клиента и предоставят пользователю стандартную статистическую информацию: среднее, минимальное, максимальное из измеренных значений, среднеквадратичное отклонение, доверительный интервал для среднего и т.д.).

Добавьте элементы сбора данных

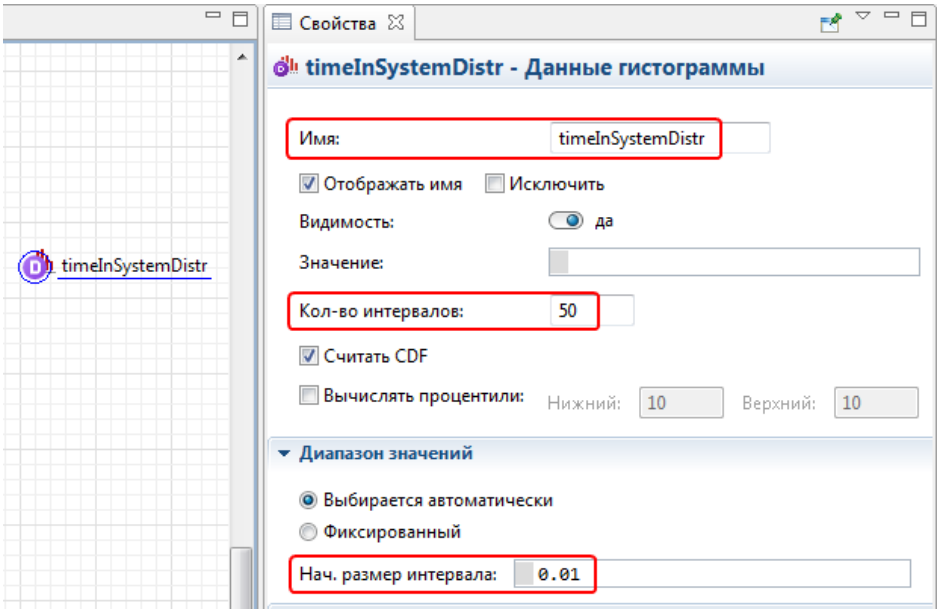
1. Чтобы добавить объект сбора данных гистограммы на диаграмму, перетащите элемент **Данные гистограммы** с палитры **Статистика** на диаграмму агента Main.



2. Задайте свойства элемента.
   - Измените **Имя** на *waitTimeDistr*.
   - Сделайте **Кол-во интервалов** равным *50*.
   - Задайте **Начальный размер интервала**: *0.01*.



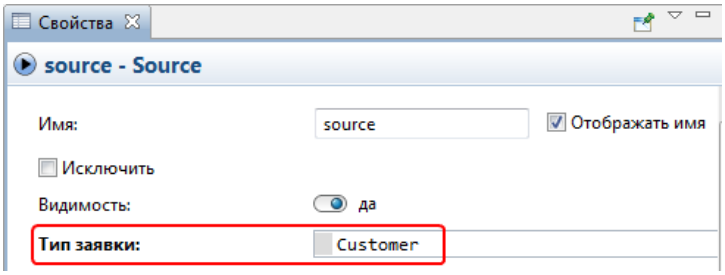
3. Создайте еще один элемент сбора данных гистограммы. Ctrl+перетащите (Mac OS: Cmd+перетащите) только что созданный объект данных гистограммы, чтобы создать его копию. Измените **Имя** этого элемента на *timeInSystemDistr*.



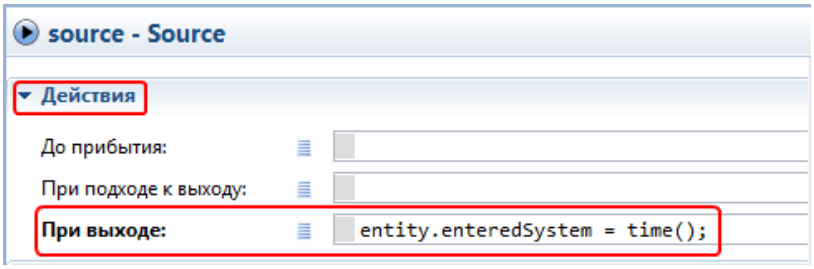
Теперь нам нужно изменить свойства блоков нашей диаграммы процесса.

Измените свойства блоков диаграммы процесса

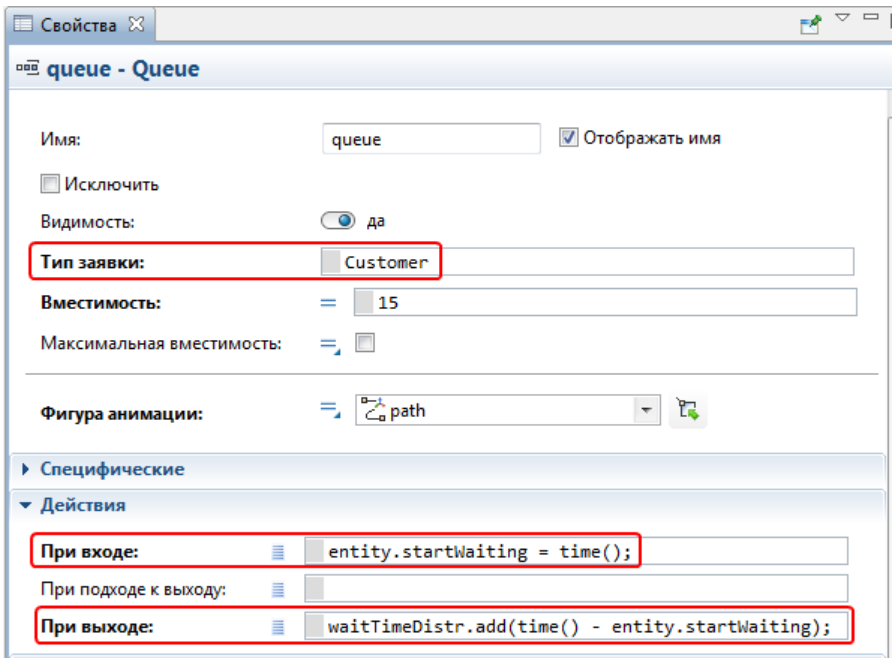
1. Измените свойства объекта *source*:
   - Введите *Customer* в поле **Тип заявки**. Это позволит напрямую обращаться к полям класса заявки Customer в коде динамических параметров этого объекта.



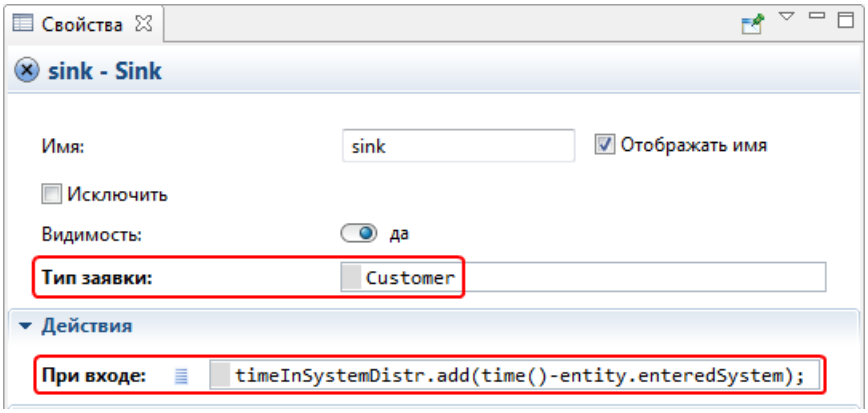
- Убедитесь, что тип заявки Customer указан в поле **Новая заявка**. Этот объект должен продолжать создавать заявки типа Customer.
- Введите entity.enteredSystem = time(); в поле действия **При выходе** в секции **Действия**. Этот код будет сохранять время создания заявки-клиента в переменной enteredSystem нашего типа заявки Customer. Функция time() возвращает текущее значение модельного времени.



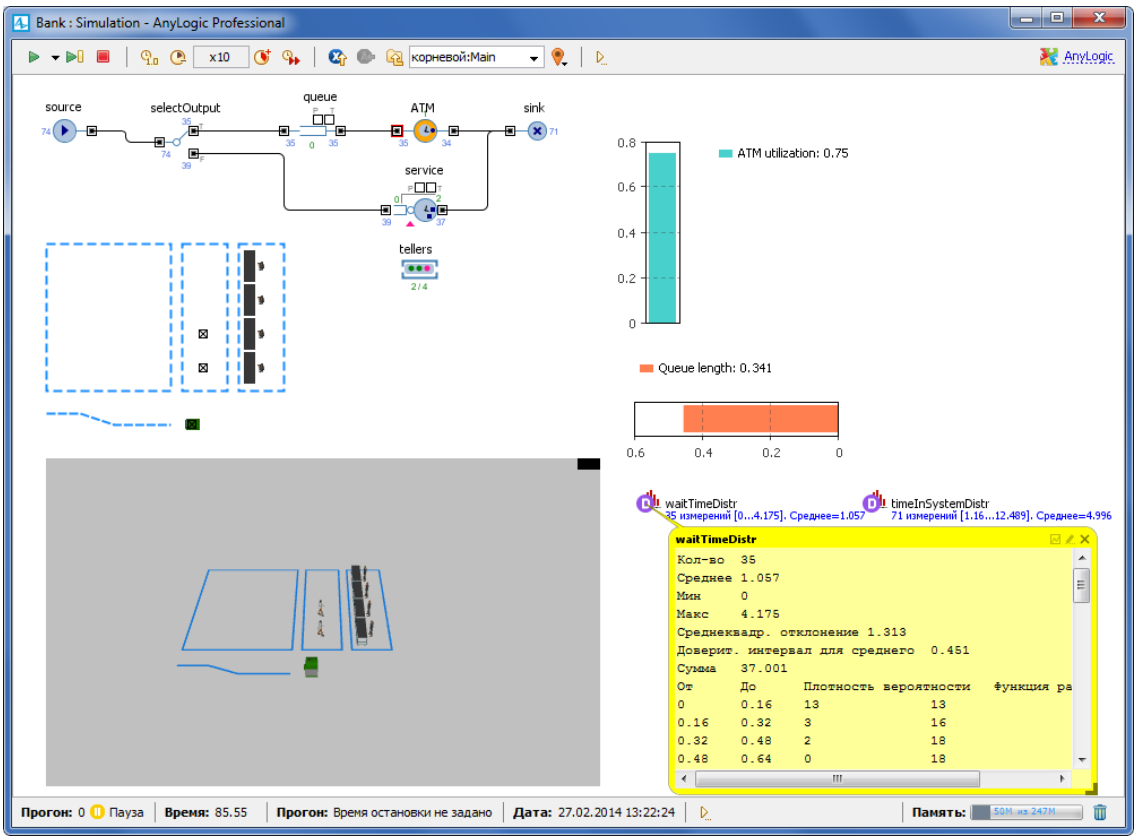
2. Измените свойства объекта *queue*:
   - Введите *Customer* в поле **Тип заявки**.
   - Введите entity.startWaiting = time(); в поле действия **При входе** в секции **Действия**. Этот код запоминает время начала ожидания клиентом его очереди на обслуживание в переменной startWaiting нашего типа заявки Customer.
   - Введите waitTimeDistr.add(time() - entity.startWaiting); в поле действия **При выходе**. Этот код добавляет время, в течение которого клиент ожидал обслуживания, в объект сбора данных *waitTimeDistr*.



3. Измените свойства объекта *sink*:
   - Введите *Customer* в поле **Тип заявки**.
   - Введите timeInSystemDistr.add(time()-entity.enteredSystem); в поле действия **При входе** в секции **Действия**. Этот код добавляет полное время пребывания клиента в банковском отделении в объект сбора данных гистограммы *timeInSystemDistr*.



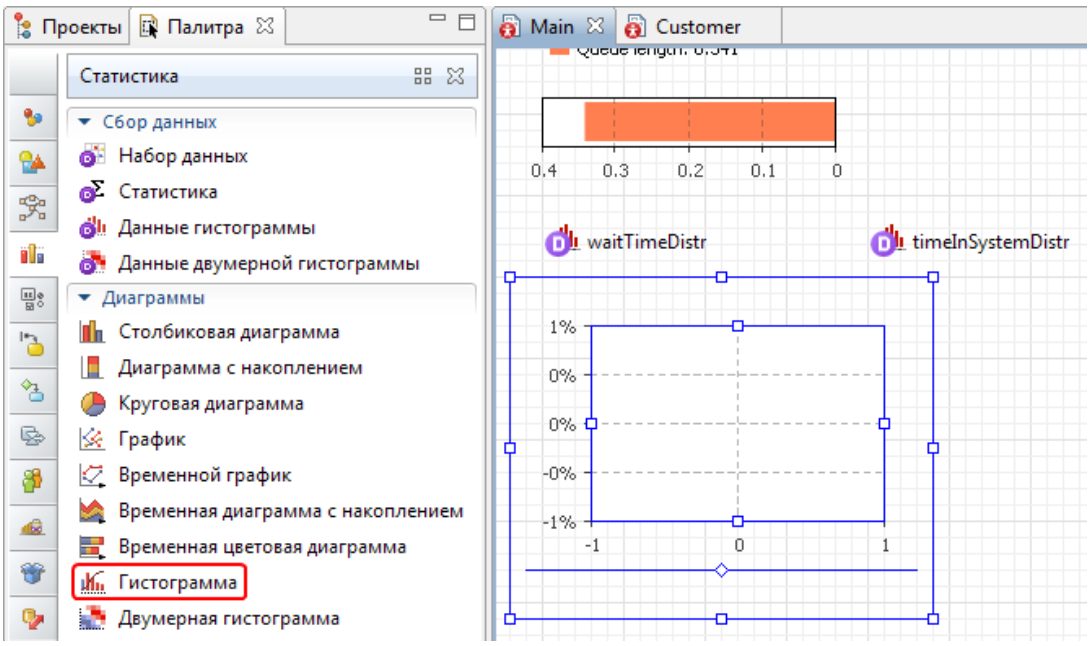
Запустите модель и просмотрите статистику с помощью окон инспекта. Открыть окно инспекта можно щелкнув мышью по значку объекта сбора данных. Здесь Вы увидите стандартные для статистического анализа данные, приведенные для значений, собранных в данном объекте сбора статистики.



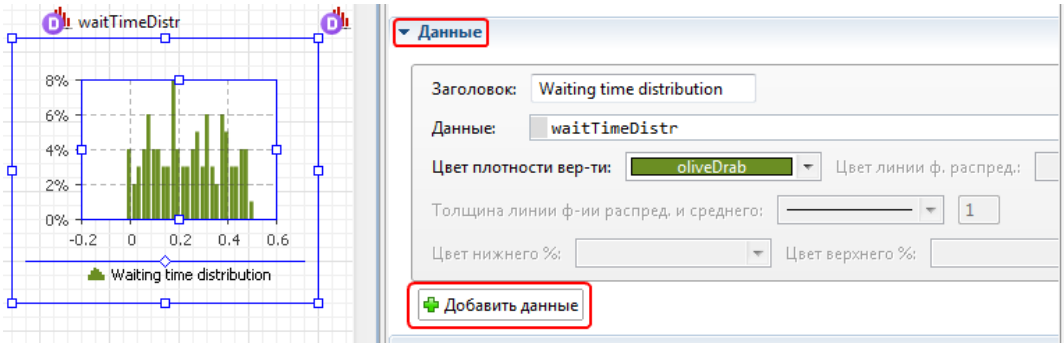
Теперь давайте добавим на диаграмму нашего типа агента гистограммы, которые будут отображать собранную нами временную статистику.

Добавьте две гистограммы для отображения распределений времен ожидания клиента и пребывания клиента в системе

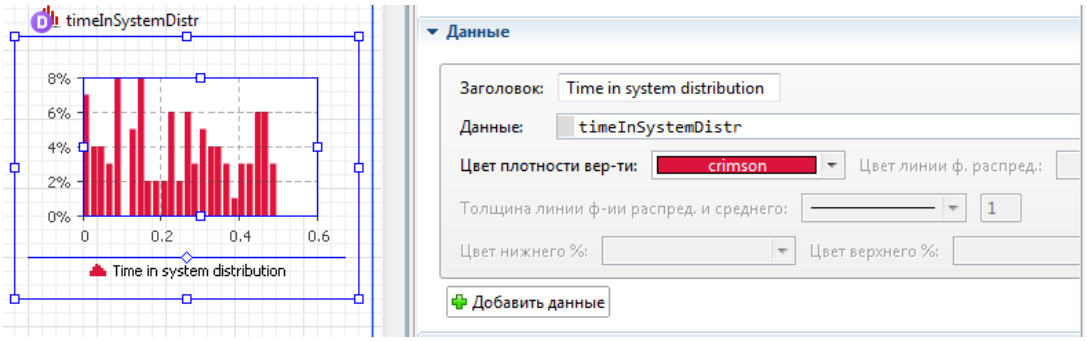
1. Чтобы добавить гистограмму на диаграмму агента, перетащите элемент **Гистограмма** из палитры **Статистика** в то место графического редактора, куда Вы хотите ее поместить. Измените ее размер при необходимости



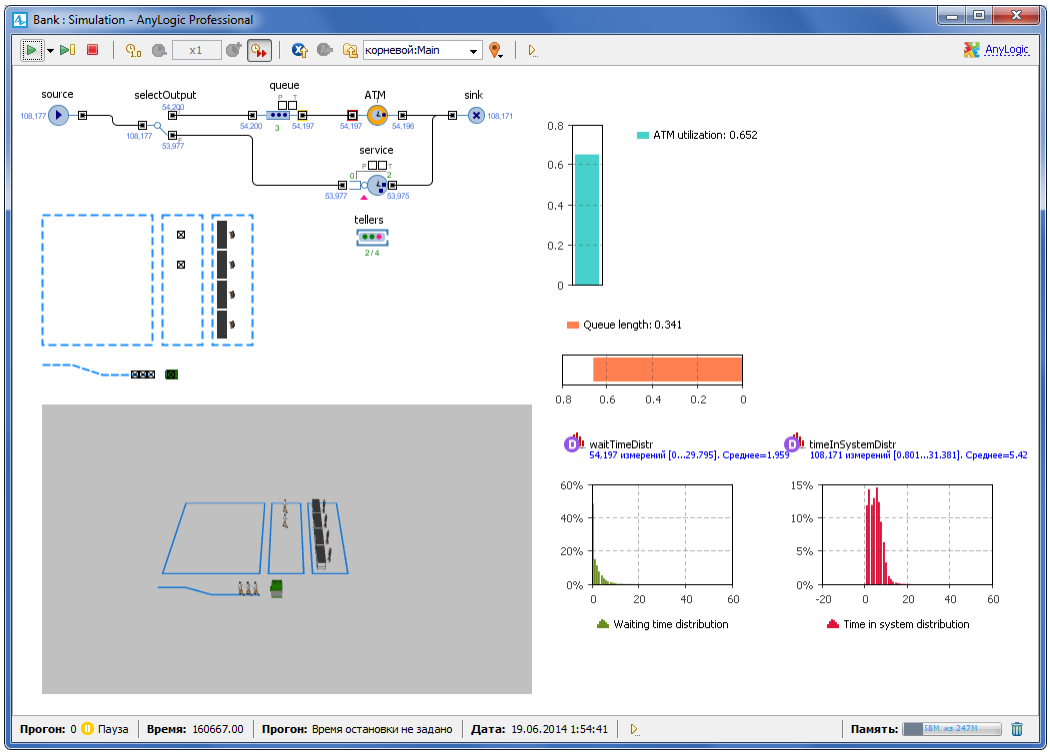
2. Укажите, какой элемент сбора данных хранит данные, которые Вы хотите отображать на гистограмме: в секции **Данные** свойств гистограммы щелкните мышью по кнопке **Добавить данные** и измените **Заголовок** отображаемых данных на *Waiting time distribution*. Введите в поле **Данные** имя соответствующего элемента: waitTimeDistr



3. Добавьте еще одну гистограмму и расположите ее под ранее добавленной.
   Измените **Заголовок** отображаемых данных на *Time in system distribution*.
   В поле **Данные** введите имя элемента, хранящего данные, которые будут отображаться на гистограмме: timeInSystemDistr.



Запустите модель. Включите режим виртуального времени и понаблюдайте за тем, какой вид примет распределение времен ожидания и пребывания клиента в системе.



Список литературы для подготовки к экзамену

1. Трахтенгерц, Э. А. Компьютерная поддержка принятия решений. Сер. «Системы и проблемы управления»/ Э. А. Трахтенгерц. – М.: СИНТЕГ, 2005. – 376 с.
2. Суздалов, Е. Г. Информационные технологии в текстильной и легкой промышленности: учеб. пособие / Е. Г. Суздалов, Т. А. Кравец, Е. Н. Якуничева, Н. Р. Туркина. – СПб.: ИПЦ СПГУТД, 2005. – 261 с.
3. Anylogic. Многоподходное имитационное моделирование. Изучаем ИМ. [эл. ресурс] // www.anylogic.ru: сервер, 2014. URL: http://www.anylogic.ru/use-of-simulation
4. Кустов, А. И. Имитационное моделирование в экономике: учеб. пособие / А. И. Кустов, В. Н. Томашевский, О. Я. Кравец. – 2-е изд., испр. – Воронеж: Научная книга, 2008. – 200 с.
5. Исследование операций в экономике: учеб. пособие для вузов/ Н. Ш. Кремер, Б.А. Путко, И. М. Тришин и др. – М.: Юнити, 2005 г, 407 с.
6. Воронов, М. В. Введение в системный анализ: учеб. пособие/ М. В. Воронов. – Тирасполь: Полиграфист, 2011. – 224 с.
7. Суздалов, Е. Г. Теория систем и системный анализ: конспект лекций/ Е. Г. Суздалов. – СПб.: СПГУТД, 2010. – 60 с.
8. Балтрашевич, В. Э., Методы оптимизации: учеб. пособие/ В. Э. Балтрашевич, Н. Е. Барабанов. – СПб.: СПбГЭТУ “ЛЭТИ”, 2001. – 80 с.
9. Глухов, В. В., Экономико- математические методы для менеджмента/ В. В. Глухов, М. Д. Медников, С. Б. Коробко. – Спб.: Лань, 2000. – 200 с.
10. Малыхин, В. И. Математика в экономике/ В. И. Малыхин. – М.: Инфра-М, 2002 – 150 с.