**Моделирование системы массового обслуживания**

**в ANYLOGIC**

**Общая информация о создании моделей**

Для создания новой модели щелкните мышью по кнопке *Создать проект*. Появится диалоговое окно, в котором вы должны будете дать имя файлу вашей модели и выбрать каталог, где он будет храниться.

В правой рабочей области отображается панель «Палитра», а внизу – панель «Свойства». Панель «Палитра» содержит разделенные по категориям элементы, которые могут быть добавлены на диаграмму класса активного объекта или эксперимента. Панель «Свойства» используется для просмотра и изменения свойств выбранного в данный момент элемента модели.

В центре рабочей области AnyLogic открывается графический редактор диаграммы класса активного объекта *Main*. Чтобы добавить объект на блок-схему модели, щелкните по объекту в окне палитры *Enterprise Library* и перетащите его мышью на структурную диаграмму. При этом его свойства будут отображены на панели «Свойства». В этом окне вы можете изменять свойства элемента в соответствии с требованиями вашей модели. Позднее для изменения свойств элемента нужно будет сначала щелчком мыши выделить его на диаграмме или в дереве проекта.

Объекты должны взаимодействовать между собой, поэтому вы должны будете соединять их друг с другом. Можно соединять объекты с помощью мыши, перетаскиванием порта одного объекта на порт другого или с помощью специального средства «Соединитель».

Модель выполняется в соответствии с набором конфигурационных установок, называемым экспериментом. Вы можете создать несколько экспериментов и изменять рабочую конфигурацию модели, просто меняя текущий эксперимент модели. Один эксперимент, названный *Simulation*, создается автоматически. Выберите его щелчком мыши по элементу дерева и измените настройки модели в окне *Свойства*.

На вкладке *Основные* можно выбрать класс, который будет запущен при запуске модели. По умолчанию в качестве корневого объекта выбран объект класса *Main*, автоматически создаваемого в каждой модели. Вы можете переименовывать классы модели. Для этого нужно выделить класс щелчком мыши по значку класса в дереве модели и затем изменить его имя в окне Свойства.

На вкладке *Модельное время* можно:

1) задать режим моделирования. В режиме реального времени задается связь модельного времени с физическим, т.е. задается количество единиц модельного времени, выполняемых в одну секунду. Режим реального времени лучше всего подходит для показа анимации. В режиме виртуального времени модель выполняется без привязки к физическому времени – она выполняется так быстро, как это возможно. Данный режим лучше всего подходит, когда требуется моделировать работу системы в течение достаточно длительного периода времени;

2) запустить модель так, чтобы она работала бесконечно, но можно и остановить ее в заданный момент времени. Вы можете остановить модель по достижении переменной заданного значения или по выполнении какого-нибудь определенного условия.

На вкладе Презентация можно определить вид и скорость выполнения прогона.

**Пример СМО в виде модели банковского отделения**

В банковском отделении находятся банкомат и стойки банковских кассиров, которые предназначены для быстрого и эффективного обслуживания посетителей банка. Операции с наличностью клиенты банка производят с помощью банкомата, а более сложные операции, такие как оплата счетов, – с помощью кассиров.

1. Создайте новую модель. Переименуйте класс *Main* в *Model*. В свойствах эксперимента *Simulation* задайте выполнение модели в режиме реального времени с выполнением одной единицы модельного времени в одну секунду.
2. Создайте блок-схему модели, которая пока будет состоять только из банкомата. Для этого перетащите в окно структуры элементы библиотеки *Enterprise Library* и соедините их так, как показано на рис. 1. Объект *source* генерирует заявки (*entities*) определенного типа через заданный временной интервал. Объект *queue* моделирует очередь клиентов, ожидающих обслуживания. Объект *delay* моделирует ОУ. Объект *sink* обозначает конец блок-схемы.

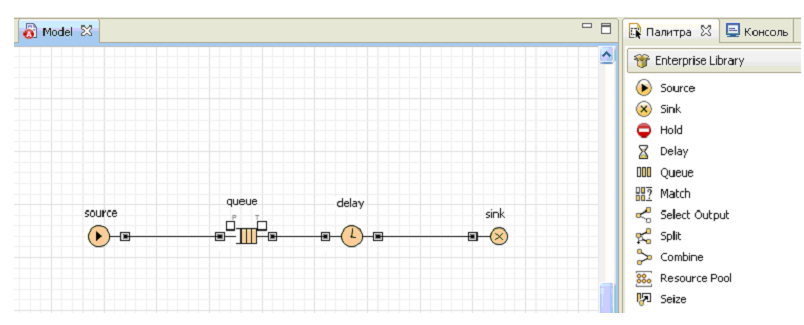


Рисунок 1.

1. Для каждой модели, созданной в *Enterprise Library*, автоматически создается блок-схема с наглядной визуализацией процесса, с помощью которой вы можете изучить текущее состояние модели, например, длину очереди, количество обслуженных человек и т.д. Для запуска модели щелкните мышью по кнопке *Запустить*. Откроется окно с презентацией запущенного эксперимента. AnyLogic автоматически помещает на презентацию каждого простого эксперимента заголовок и кнопку, позволяющую запустить модель и перейти на презентацию, нарисованную вами для главного класса активного объекта этого эксперимента.
2. Задайте данные модели, изменяя свойства созданных объектов. В свойстве *interarrivalTime* объекта *source* укажите, как часто в отделение приходят клиенты – exponential(0.67). Интервал между приходом клиентов распределен экспоненциально со средним значением, равным 1.5 единицы модельного времени. Заметьте, что аргумент функции exponential() равен 0.67, потому что в качестве аргумента задается интенсивность прихода клиентов. В свойстве *capacity* объекта *queue* задайте максимальную длину очереди – 15. В свойстве *delayTime* объекта *delay* задайте время время обслуживания – triangular( 0.8, 1, 1.3 ). Здесь время обслуживания распределено по треугольному закону со средним значением, равным 1 минуте, минимальным – 0.8 и максимальным – 1.3 минуты.
3. AnyLogic позволяет производить сбор сложной статистики. Для этого нужно лишь включить у объекта режим сбора статистики, поскольку по умолчанию он отключен для повышения скорости выполнения модели. В системе собирается статистика по длине очереди для блока *queue* (*length*) и статистика по коэффициенту использования для блока *delay* (*utilization*). Чтобы включить сбор статистики для объекта, установите переключатель **Включить сбор статистики** на вкладке **Основные** свойств объекта.
4. Усложните модель:

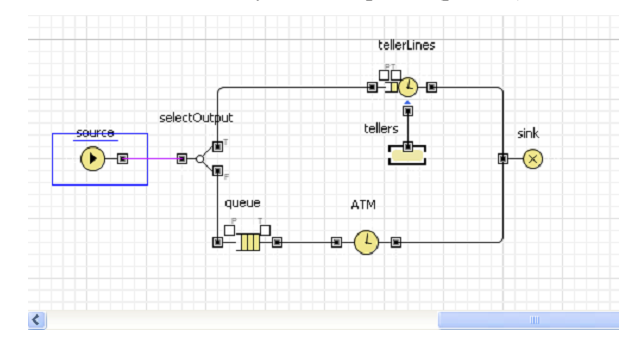
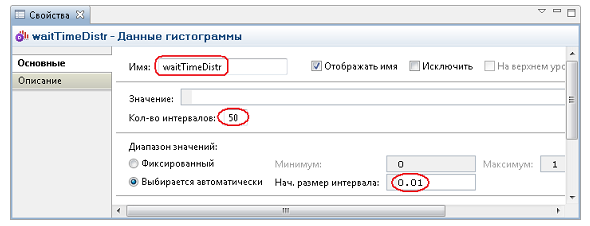


Рисунок 2.

1. *selectOutput* – является блоком принятия решения. В зависимости от заданного вами условия, заявка, поступившая в этот объект, будет поступать на один из двух выходов объекта. Оставьте свойство *selectCondition* – **uniform() < 0.5**, тогда к кассирам и банкомату будет приходить примерно равное количество клиентов;
2. *tellerLines* – моделирует многоканальное обслуживающее устройство + накопитель для этого ОУ. Число каналов задается объектом ресурсов (пункт 9 ниже). Укажите, что в очереди к кассирам может находиться до 20 человек (свойство **queueCapacity**); задайте время обслуживания (свойство **delayTime**). Будем полагать, что время обслуживания имеет треугольное распределение с минимальным средним значением 2.5, средним – 6 и максимальным – 11;
3. *ResourcePool* – задает ресурсы определенного типа. Он должен быть подсоединен к объектам, моделирующим занятие и освобождение ресурсов (в нашем случае это объект *tellerLines*). Задайте следующие свойства объекта: назовите объект *tellers*; задайте число кассиров (свойство **capacity**) – 4.
4. Необходимо определить, сколько времени клиент проводит в банковском отделении и сколько времени он теряет, ожидая своей очереди. Соберем эту статистику с помощью специальных объектов сбора данных и отобразим собранную статистику распределения времени обслуживания клиентов с помощью гистограмм. Создадим новый агент *Customer*. Сообщения этого класса будут представлять клиентов банковского отделения.
5. В панели **Проект**, щелкните правой кнопкой мыши по элементу модели и выберите **Создать** | **Agent класс** из контекстного меню. Появится диалоговое окно **Новый класс**. В поле **Имя** введите имя нового класса *Customer*;
6. На поле нового агента, подобно обычной модели, можно задать **параметры**. Перенесите из палитры следующие параметры: *enteredSystem* для сохранения момента времени, когда клиент пришел в банковское отделение; *startWaiting* для сохранения момента времени, когда клиент встал в очередь к банкомату;
7. Теперь вычислим время, которое тратится персоналом банка на обслуживание клиентов, и время, которое клиенты тратят на ожидание своей очереди. Перетащите элемент **Данные гистограммы** с палитры **Статистика** на диаграмму активного класса; задайте свойства элемента



1. Создайте еще один элемент сбора данных гистограммы *timeInSystemDistr*.
2. Измените свойства блоков вашей диаграммы процесса. Задайте следующие свойства объектов диаграммы:
3. Блок *source*, свойство **Агент** – выберите *Customer()*. Это позволит напрямую обращаться к полям класса заявки *Customer* в коде динамических параметров этого объекта. Введите *agent.enteredSystem = time();* в поле **Действие при выходе**. Этот код будет сохранять время создания заявки-клиента в переменной *enteredSystem* нашего класса заявки *Customer*. Функция time() возвращает текущее значение модельного времени;
4. Блок *tellerLines* – введите *Customer* в поле **Класс заявки**. Добавьте код в поля: **Действие при входе**: *agent.startWaiting = time();*

**Действие при выходе**: *waitTimeDistr.add(time() - agent.startWaiting);*

1. Блок *queue* – введите *Customer* в поле **Класс заявки**. Добавьте код в поля:

**Действие при входе**: *agent.startWaiting = time();*

**Действие при выходе**: *waitTimeDistr.add(time()-agent.startWaiting);*

1. Блок *ATM* (блок *delay*) – введите *Customer* в поле **Класс заявки**;
2. Блок *sink* – введите *Customer* в поле **Класс заявки**. Напишите следующий код, чтобы сохранить в наборах данных данные о клиенте, покидающем банковское отделение (**Действие при входе**): *timeInSystemDistr.add(time()-agent.enteredSystem);*
3. Добавьте две гистограммы для отображения распределений времен ожидания клиента и пребывания клиента в системе. Чтобы добавить гистограмму на диаграмму класса активного объекта, перетащите элемент **Гистограмма** из палитры **Статистика** в то место графического редактора, куда вы хотите ее поместить. Укажите, какой элемент сбора данных хранит данные, которые хотите отображать на гистограмме: щелкните мышью по кнопке **Добавить данные** и введите в поле **Данные** имя соответствующего элемента – *waitTimeDistr*. Аналогичным образом добавьте еще одну гистограмму и расположите ее под ранее добавленной. В поле **Данные** введите *timeInSystemDistr*.
4. Запустите модель и оцените результаты ее работы.
5. Измените модель в соответствии со своим вариантом из лабораторной работы №1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант, № | Распределение входное | Вероятность обращения к кассиру/банкомату | Время обслуживания клиента | Количество кассиров |
| 1 | Экспоненциальное | 1/1 | 4±2 | 2 |
| 2 | Экспоненциальное | 1/2 | 6±2 | 3 |
| 3 | Экспоненциальное | 2/1 | 8±2 | 4 |
| 4 | Равномерное | 1/3 | 7±2 | 5 |
| 5 | Экспоненциальное | 3/1 | 9±2 | 6 |
| 6 | Экспоненциальное | 1/1 | 4±3 | 6 |
| 7 | Равномерное | 1/2 | 6±3 | 5 |
| 8 | Равномерное | 2/1 | 8±3 | 4 |
| 9 | Экспоненциальное | 1/3 | 7±3 | 3 |
| 10 | Экспоненциальное | 3/1 | 9±3 | 2 |
| 11 | Равномерное | 1/1 | 4±1 | 2 |
| 12 | Экспоненциальное | 1/2 | 6±1 | 3 |
| 13 | Экспоненциальное | 2/1 | 8±1 | 4 |
| 14 | Равномерное | 1/3 | 7±1 | 5 |
| 15 | Равномерное | 3/1 | 9±1 | 6 |
| 16 | Экспоненциальное | 1/1 | 4±2 | 6 |
| 17 | Экспоненциальное | 1/2 | 6±2 | 5 |
| 18 | Равномерное | 2/1 | 8±2 | 4 |
| 19 | Равномерное | 1/3 | 7±2 | 3 |
| 20 | Экспоненциальное | 3/1 | 9±2 | 2 |