**Лабораторная работа №4**

**Агентное моделирование**

**Цель работы:**

1. Изучить агентный подход в моделировании сложных систем.
2. Изучить возможности AnyLogic для создания агентных моделей.

**Практическое задание**

1. **Постановка задачи**

В ходе работы создается классическая модель распространения инноваций Басса и те ее расширения, которые демонстрируют возможности AnyLogic для создания агентных моделей.

*Агент* – это некоторая сущность, которая обладает активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, может взаимодействовать с окружением и другими агентами, а также может изменяться (эволюционировать). Многоагентные (или просто агентные) модели используются для исследования *децентрализованных систем*, динамика функционирования которых определяется не глобальными правилами и законами, а, наоборот, эти глобальные правила и законы являются результатом индивидуальной деятельности членов группы. Цель агентных моделей – получить представление об общем поведении системы исходя из знаний о поведении ее отдельных активных объектов и взаимодействии этих объектов в системе. Агентная модель может содержать десятки и даже сотни тысяч активных агентов (это могут быть люди, компании, активы, проекты, транспортные средства, города, животные и т.д.). Установив возможные связи, определяя их поведение в некой окружающей среде, можно изучить индивидуальное поведение каждого агента образующего глобальное поведение моделируемой системы. При помощи агентов моделируют рынки (агент – потенциальный покупатель), конкуренцию и цепочки поставок (агент – компания), население (агент – семья, житель города или избиратель) и многое другое.

В среде AnyLogic с помощью параметров можно задавать пол агента, дату рождения, его координаты в пространстве, социальные свойства. Стейтчарты и таймеры могут выражать поведение: состояния агента и изменение состояний под воздействием событий и условий. Например, переходы в разные возрастные или социальные группы, изменения образования или дохода и т.д. Кроме того, агент может иметь интерфейс для взаимодействия с окружением.

1. **Построение модели**
2. Создайте модель с именем Mag. Задайте единицы модельного времени – годы.
3. Создайте популяцию агентов. Для этого:

* Перетащите элемент **Агент** http://127.0.0.1:60324/help/topic/com.xj.anylogic.help/html/agentbased/images/Agent_icon.gif из палитры **Основная** на графическую диаграмму (рис 4.1). Вы увидите окно **Мастера создания агентов**: **Создание агентов**. Щелкните кнопку **Популяция агентов**(рис.4.2)

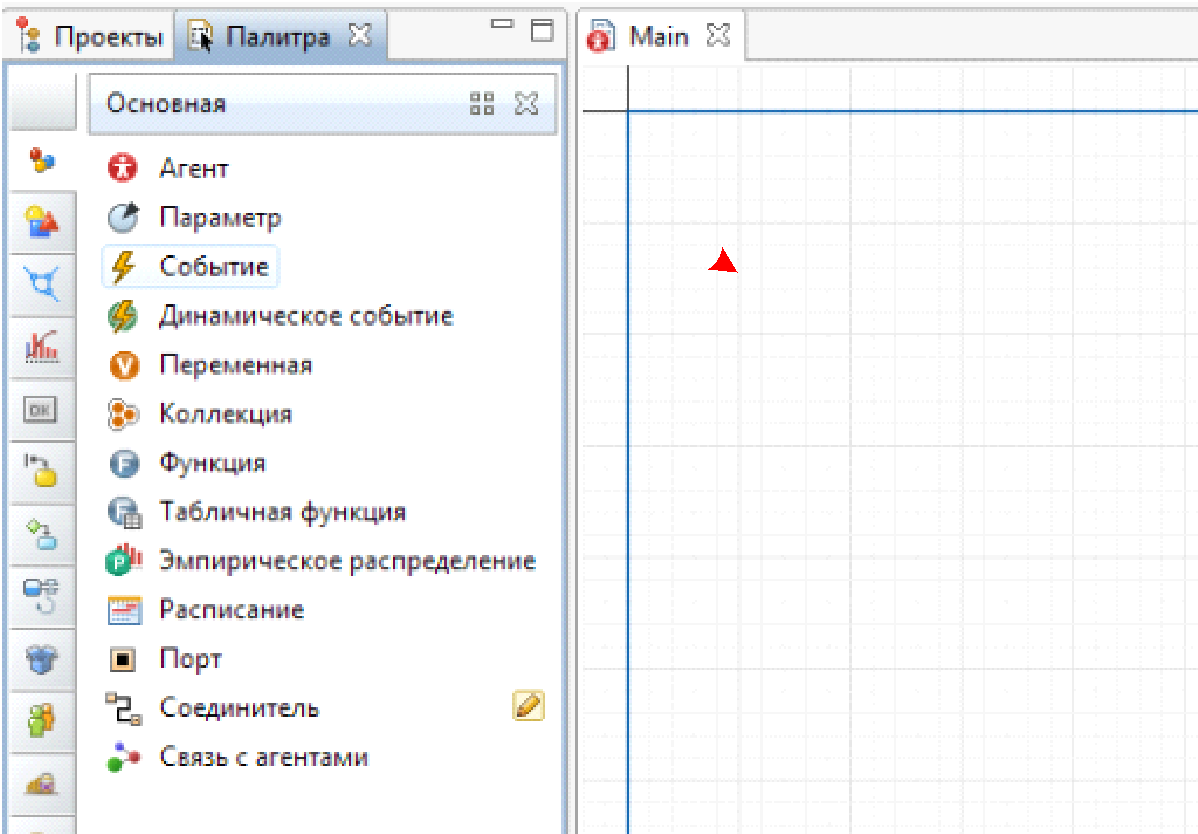


Рис. 4.1

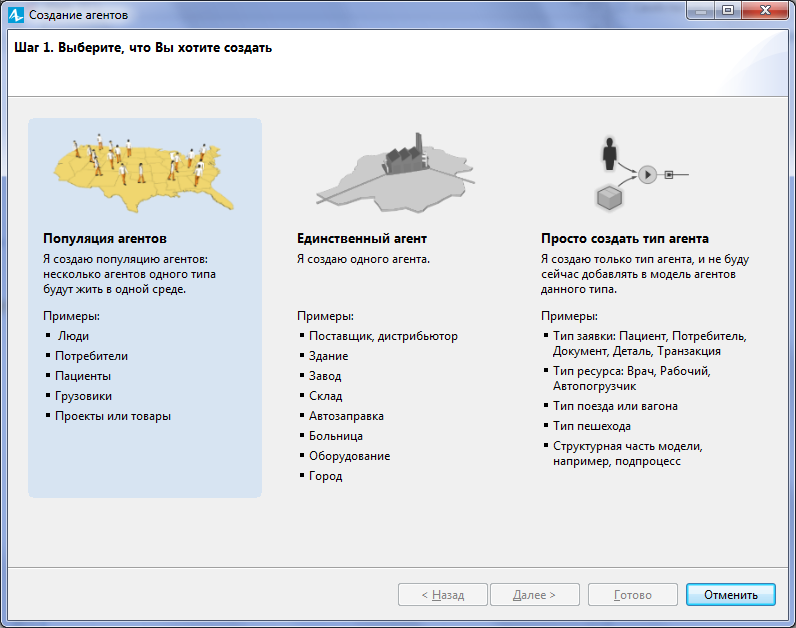


Рис. 4.2

* На следующем шаге **Мастера** (рис.4.3) откажитесь от использования шаблонов.

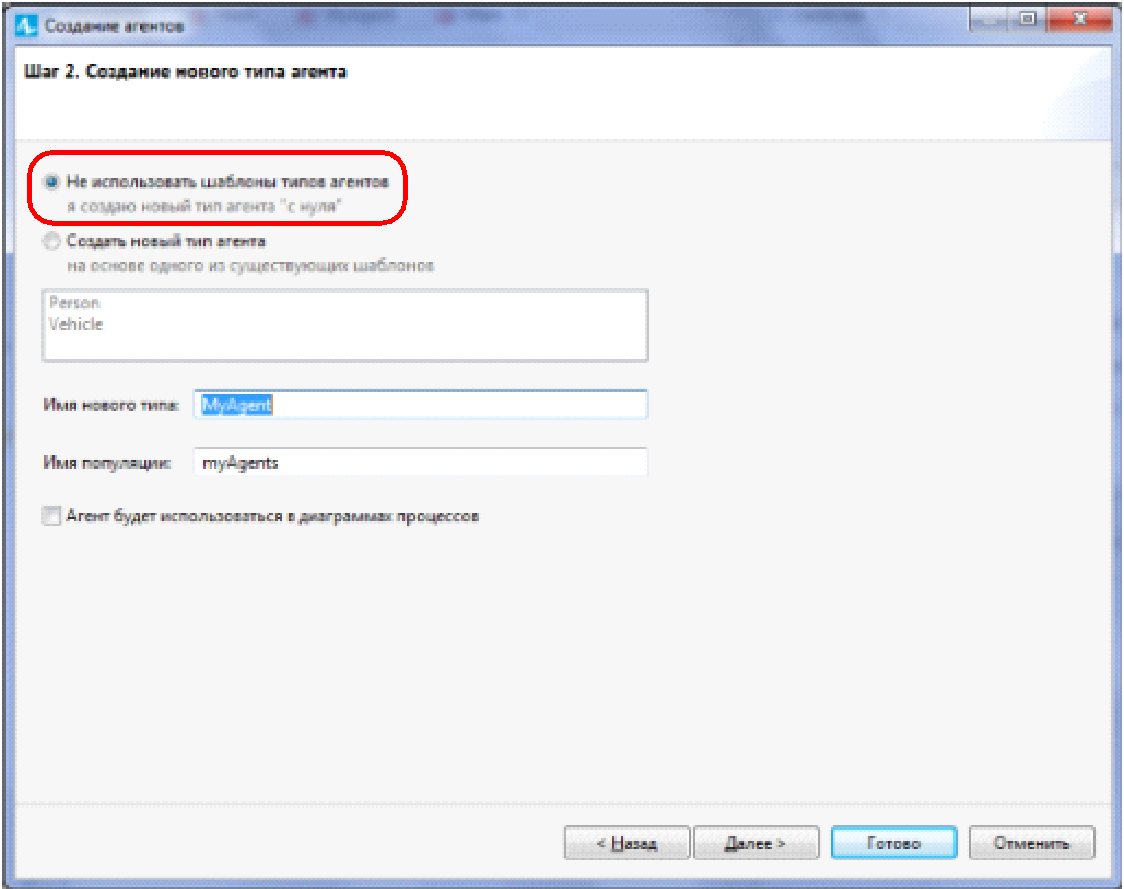


Рис. 4.3

* На третьем шаге **Мастера** выберите тип анимации агента – **Человек,**  опция **2D** (рис. 4.4). Опция **3D** позволит отображать агенты и в 2D, и в 3D (опция **Нет** выбирается, если ни одна фигура анимации не подходит).

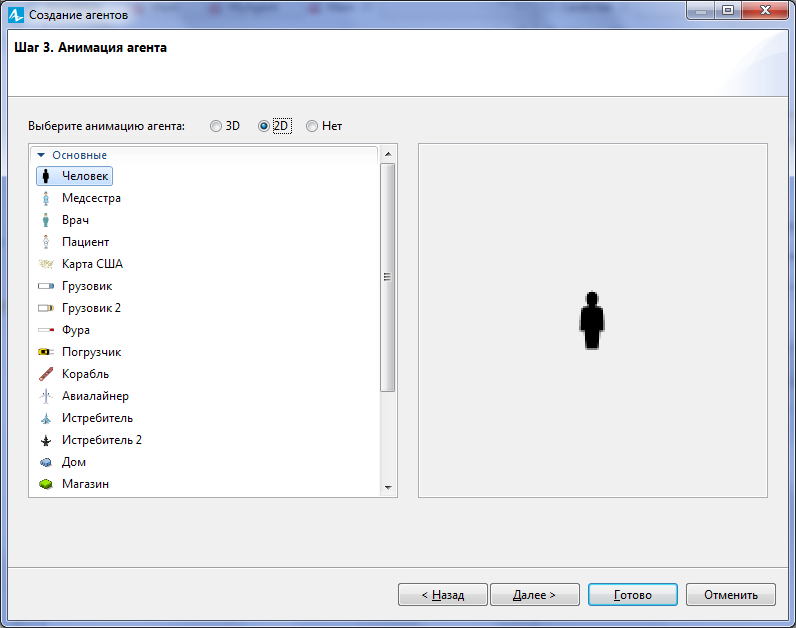


Рис. 4.4

* Четвертый шаг **Мастера** (рис. 4.5) позволяет установить параметры для этого типа агентов. Щелкните **< добавить... >** , чтобы создать новый параметр AdEffectiveness, задающий подверженность человека влиянию рекламы. Выделите щелчком мыши параметр и укажите его имя, тип (double) на панели настроек справа.

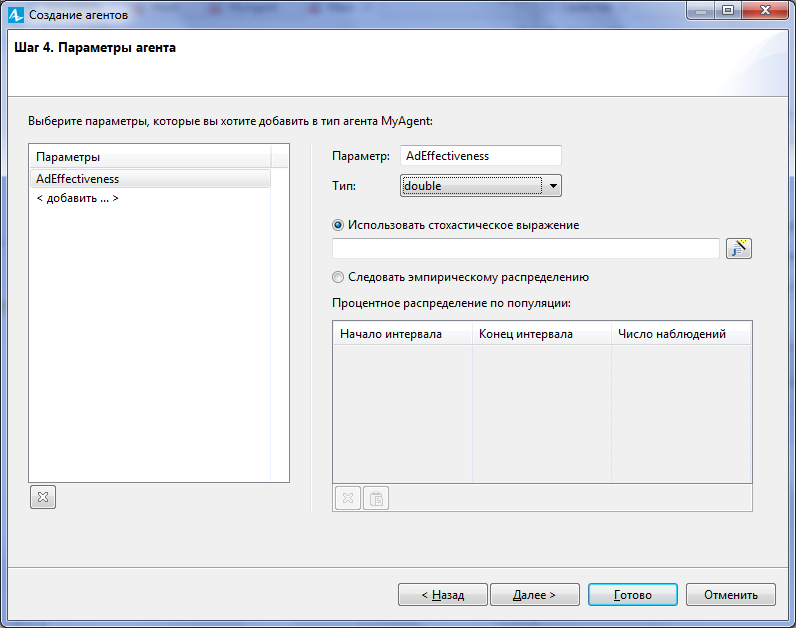


Рис. 4.5

* Укажите количество агентов, которое будет изначально создано в модели – 1000 (рис. 4.6)

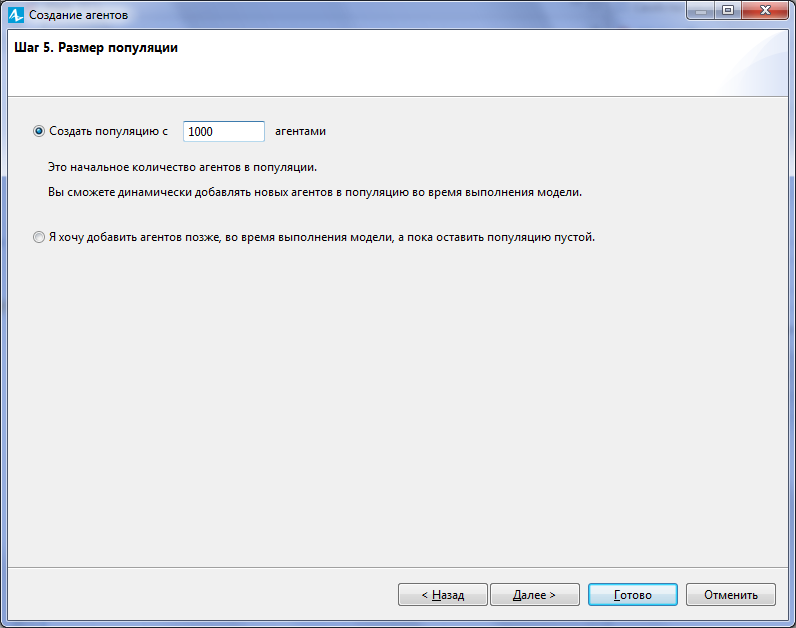


Рис. 4.6

* Задайте тип пространства, в котором будут обитать агенты – **Непрерывное** (рис.4.7). Задайте размерности данного пространства: введите в поле **Ширина** – 600, а в поле **Высота** – 350. В результате агенты будут располагаться в пределах непрерывного пространства, отображаемого на презентации модели областью размером 600x350 пикселей. Агенты могут обитать в различных типах пространств. В непрерывном пространстве есть возможность изменять местоположение агента и получать информацию о его текущем местоположении, перемещать агента с заданной скоростью из одного места в другое, выполнять действия по его прибытию в место назначения, создавать анимацию (статического или движущегося) агента, устанавливать соединения согласно выбранному шаблону расположения агентов и многие другие возможности.

Задайте тип сети взаимосвязей агентов –**Случайное .**  Количество связей у агента 2 (рис. 4.8). Нажмите кнопку **Готово.**

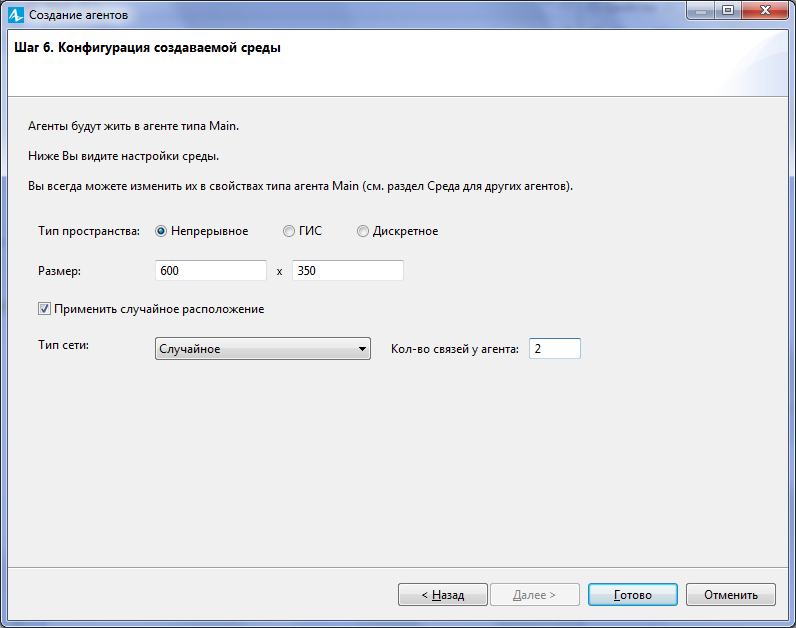


Рис. 4.7

**Стандартные сети агентов**

Взаимодействие агентов в агентных моделях может быть реализовано различными способами. Если связи между агентами достаточно постоянны, то агенту нужно запоминать тех агентов, которые состоят с ним в какой-либо связи. Смысл таких связей может быть, например, следующим: друг, коллега, родитель, ребенок и т.д.

Списки взаимодействий используются, когда агент посылает сообщения своим контактам.

AnyLogic предоставляет встроенный механизм поддержки "плоских" связей, таких, как друзья, социальные контакты и т.д. Любой агент (активный объект, объявленный **Агентом**) может иметь определенное число связей - ссылок на других агентов, обитающих в той же среде.

AnyLogic поддерживает несколько типов сетей агентов:

*Случайное* (Agent.NETWORK\_RANDOM) - агенты соединяются случайно, у каждого агента устанавливается заданное количество связей

*Согласно расстоянию* (Agent.NETWORK\_ALL\_IN\_RANGE) - друг с другом соединяются те агенты, расстояние между которыми не больше заданного  радиуса соединения (только в непрерывном пространстве)

*Решеточно упорядоченное кольцо* (Agent.NETWORK\_RING\_LATTICE) - связи агентов образуют кольцо, в котором каждый агент соединяется с заданным количеством ближайших агентов.

*Малый мир* (Agent.NETWORK\_SMALL\_WORLD) - представляет собой решеточно упорядоченное кольцо, где некоторые связи были разорваны и установлены с удаленными агентами.

*Безразмерная* (Agent.NETWORK\_SCALE\_FREE) - некоторые агенты являются "хабами" (или концентраторами) с множествами соединений, а некоторые -  and some are "отшельниками" с небольшим числом соединений

1. Откройте структурную диаграмму класса *MyAgent*. Обратите внимание на неудаляемый элемент http://127.0.0.1:49752/help/topic/com.xj.anylogic.help/html/agentbased/images/AgentLink_obj.gif*connections*, задающий настройки взаимодействия агента в среде. В созданном **Мастером** параметре *AdEffectiveness* в окне его свойств задайте значение по умолчанию 0.011.

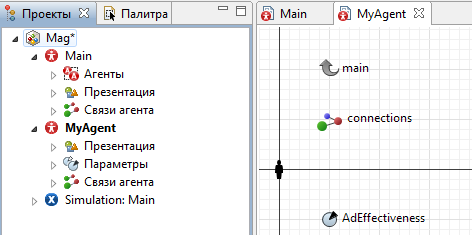


Рис. 4.8.

1. Задайте поведение агента. Поведение агента обычно описывается в классе этого агента (в данном случае это класс *MyAgent*) с помощью *диаграммы состояний.* Создайте диаграмму состояний представленную на рисунке 4.9

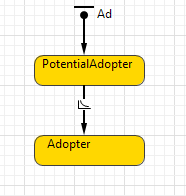


Рис.4.9

Первое состояние *PotentialAdopter -* начальное состояние, оно означает, что человек еще не купил продукт. Нижнее состояние *Adopter* означает, что человек уже купил продукт. Переход из состояния *PotentialAdopter* в состояние *Adopter* будет моделировать покупку продукта. В окне свойств перехода выберите **С заданной интенсивностью** из выпадающего списка **Происходит** и введите *AdEffectiveness* в расположенном ниже поле **Интенсивность**. Время, через которое человек купит продукт, экспоненциально зависит от эффективности рекламы продукта.

1. На структурной диаграмме класса *MyAgent* выделите элемент **Человек** и щелкните по нему правой кнопкой мыши , выберите из списка **Выделить содержимое группы** (Рис. 4.10) .

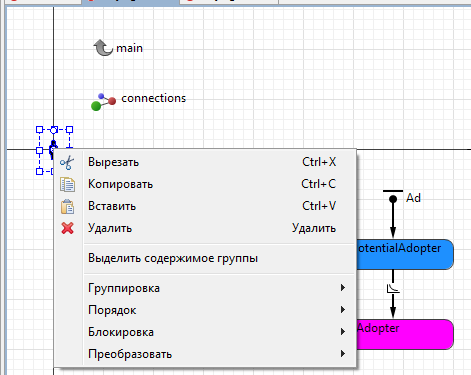


Рис. 4.10

Базовые картинки AnyLogic представляют собой не растровые изображения, а группы, состоящие из обычных фигур презентации . При необходимости можно самостоятельно изменить любую из фигур, с помощью которых была нарисована эта картинка, если она по каким-то параметрам (размер, цвет и т. д.) не будет удовлетворять представлениям пользователя о ней. Делается это с помощью динамических свойств группы фигур. Введем выражение Ad.isStateActive(Adopter)?red :blue задающее **Цвет заливки** элемента **Челове**к в зависимости от того, какое состояние стейтчарта является активным (рис.4.11) То есть купившие продукт будут отображаться красным цветом, а потенциальные покупатели –синим.

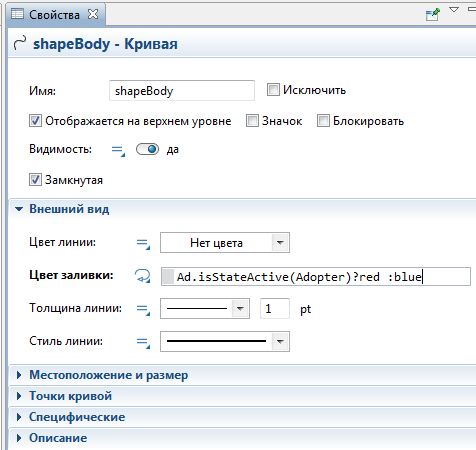


Рис. 4.11

1. Настройте выполнение модели. В окне свойств эксперимента перейдите на вкладку **Модельное время** и задайте останов модели после 8 единиц модельного времени (Рис.4.12).

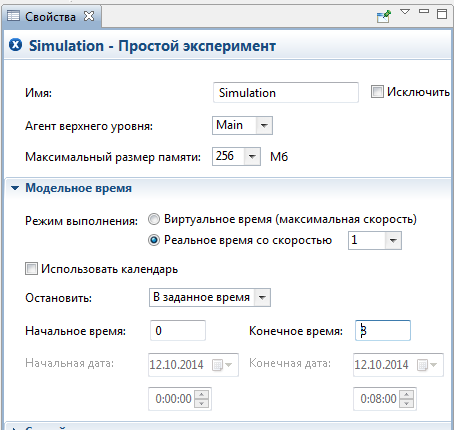


Рис. 4.12

1. Постройте проект с помощью кнопки панели инструментов **Построить** (клавиша **F7**). Если ошибок в проекте нет, то запустите модель. Результат представлен на рисунке 4.13.

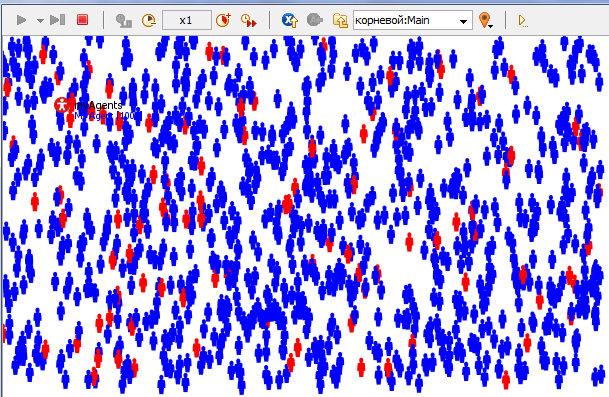


Рис. 4.13

8. Произведем подсчет потребителей продукта. Главная задача модели распространения продукта – изучение того, как быстро люди покупают новый продукт. Для этого будем подсчитывать число потребителей и потенциальных потребителей продукта, что можно сделать с помощью функций сбора статистики.

* Откройте диаграмму класса *Main*. Выделите на диаграмме вложенный объект *myAgent*.
* Перейдите на вкладку **Статистика** панели свойств объекта *myAgent.* Щелкните мышью по кнопке **Добавить функцию сбора статистики**. Откроется секция свойств для задания свойств новой функции сбора статистики по элементам этого реплицированного объекта (*myAgent*).
* Задайте имя функции – *potentialAdopters*. Оставьте выбранный по умолчанию **Тип функции** – кол-во. Задайте **Условие**: item.Ad.isStateActive(item.PotentialAdopter). Эта функция будет вести подсчет количества агентов, для которых выполняется заданное условие, т.e. тех агентов, которые находятся в текущий момент времени в состоянии *PotentialAdopter* (являются потенциальными потребителями продукта). Текущий элемент реплицированного объекта (т.е. тот, по которому в текущий момент проходит итерационный цикл функции сбора статистики) доступен в этих полях как item.
* Создайте еще одну функцию сбора статистики (рис. 4.14 ). Назовите ее *adopters*. **Тип функции** – кол-во. Условие: item.Ad.isStateActive(item.Adopter) Данная функция будет вести подсчет количества агентов, которые находятся в состоянии *Adopter* (т.е. уже приобрели продукт).

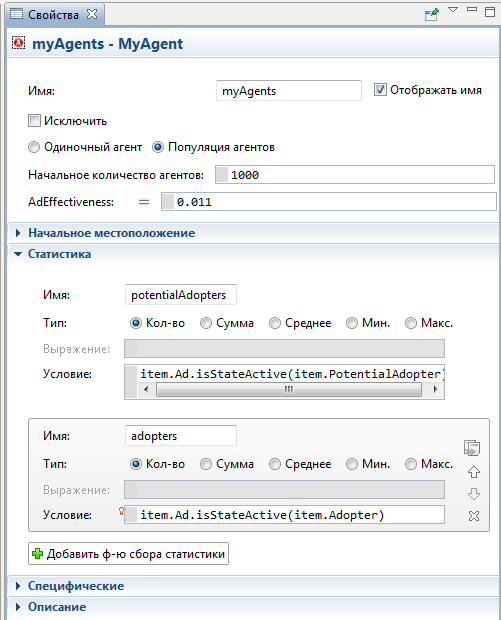


Рис. 4.14

1. Добавьте временной график, отображающий динамику изменения численности потребителей и потенциальных потребителей продукта. Расположите его, как показано на рис. 4.15 .

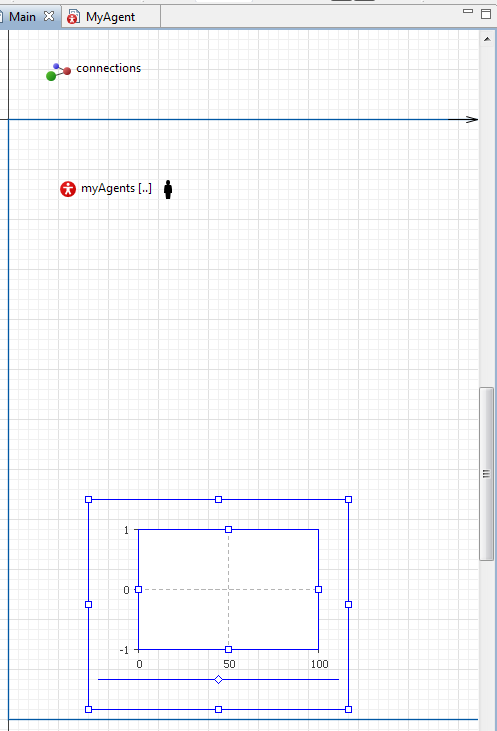


Рис. 4.15

Настройте свойства графика (рис. 4.16).

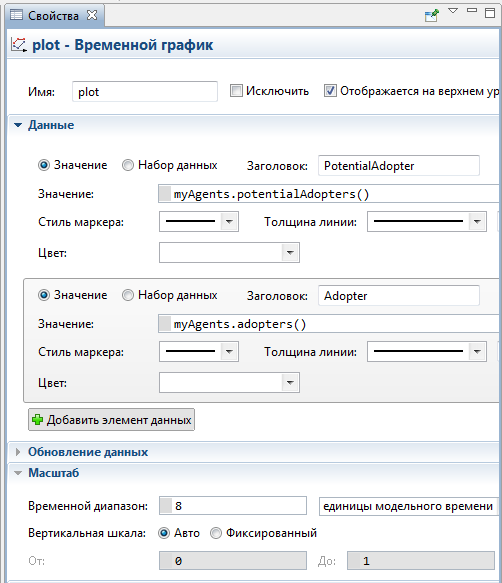


Рис. 4.15

1. Запустите модель. На графике (рис. 4.16 ) просмотрите динамику моделируемого процесса. Вы увидите, что под влиянием рекламы каждую единицу времени постоянная доля от общей численности потенциальных потребителей продукта приобретает изучаемый нами продукт.

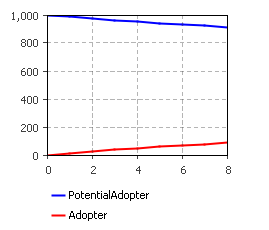


Рис. 4.16

**Учет влияния общения людей**

В текущей модели люди приобретают продукт только под влиянием рекламы. На самом деле рекламный эффект играет значительную роль только в момент выпуска продукта на рынок. В дальнейшем все большую роль будет играть общение людей со своими знакомыми, которые этот продукт уже приобрели. Чтобы учесть влияние общения людей, внесем в модель небольшие изменения.

1. Откройте диаграмму класса *MyAgent* и создайте два новых параметра: параметр *ContactRate* – среднегодовое количество встреч человека (предположим, что человек в среднем встречается со 100 людьми в год), тогда значение по умолчанию 100, тип - *int* ; параметр *AdoptionFraction* – сила убеждения человека, влияющая на то, сколько людей он сможет убедить в необходимости купить продукт. Значение по умолчанию – 0.015. Тип – *double*.
2. Измените стейтчарт агента:

* Откройте диаграмму стейтчарта, добавьте в состояние *Adopter* внутренний переход (рис. 4.17 ). Для этого щелкните мышью по кнопке панели инструментов **Переход**, затем поочередно щелкните по любым двум сторонам состояния *Adopter*;

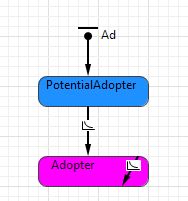
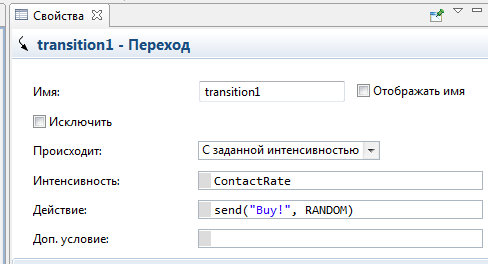


Рис. 4.17

* Задайте интенсивность, по которой происходит переход – ContactRate. Данный переход будет моделировать покупку продукта знакомым этого человека. Насколько быстро владелец продукта убедит своего знакомого в необходимости покупки зависит от силы убеждения этого человека и от того, сколько знакомых он встречает за год;
* Задайте **Действие** перехода: send("Buy!", RANDOM); Метод *send()* отсылает сообщение другому агенту. Первый аргумент задает сообщение, которое будет послано, а второй задает агента, которому сообщение будет адресовано. В нашем случае сообщение посылается какому-то случайно выбранному агенту, поэтому в качестве значения аргумента используется специальная константа RANDOM (рис. 4.18 ). Данный переход генерирует сигнал для какого-то знакомого, затем срабатывает переход стейтчарта, моделирующий покупку продукта этим знакомым;



* Рис. 4. 18
* Добавьте еще один переход из состояния *PotentialAdopter* в состояние *Adopter* (рис. 4.19). Он будет срабатывать по сигналу, который будет генерироваться внутренним переходом состояния *Adopter*;

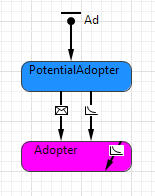


Рис. 4.19

Измените свойства этого перехода. Вероятность принятия решения о покупке продукта будет зависеть от силы убеждения человека. В нашей модели данная характеристика задается параметром *AdoptionFraction*. Введите randomTrue(AdoptionFraction) в поле **Доп. условие** (**randomTrue** генерирует истинное значение (true) с заданной вероятностью p).

В результате введения дополнительного условия продукт будет приобретаться с вероятностью, задаваемой параметром *AdoptionFraction*. Переход будет срабатывать, когда диаграмма состояний этого агента получит сообщение "Buy!" («Купи!») от другого агента – своего знакомого. Чтобы этот переход срабатывал при получении сообщения, на странице свойств данного перехода выберите из выпадающего списка **Происходит** *При получении сообщения* (рис. 4.20). Укажите, что переход будет срабатывать только при по-лучении сообщения соответствующего содержания. Для этого выберите из группы **Тип сообщения** опцию **String**, далее – опцию **Если сообщение равно** и введите "Buy!" в расположенном ниже поле.

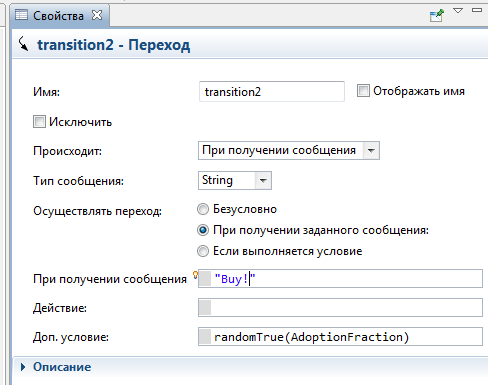


Рис. 4.20

* Измените свойства агента. На диаграмме класса *MyAgent* (рис. 4.21), в таблице свойств элемента http://127.0.0.1:49752/help/topic/com.xj.anylogic.help/html/agentbased/images/AgentLink_obj.gif*connections*, в поле **Действие при получении** сообщения введите Ad.receiveMessage(msg); При поступлении сообщения в диаграмму состояний, вызывается метод receiveMessage() этой диаграммы состояний.

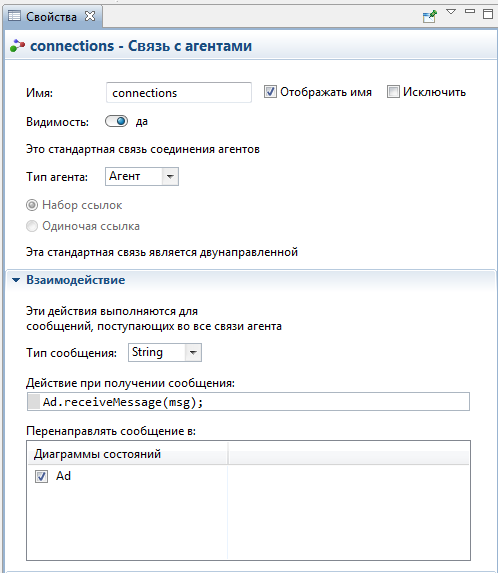


Рис. 4.21

Запустите модель. Изучите динамику изменения числа потребителей и потенциальных потребителей продукта. Графики переменных должны представлять собой *S*-образные кривые.

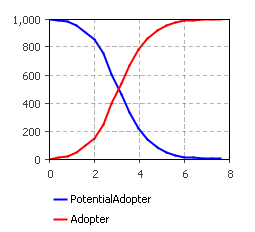


Рис. 4.22