



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Государственное образовательное учреждение
высшего профессионального образования**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПЕЧАТИ ИМЕНИ ИВАНА ФЕДОРОВА**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**к выполнению лабораторных работ
по дисциплине «Моделирование систем»**

для специальности 220200.65 – Автоматизированные системы
обработки информации и управления
и по направлению

230100.62 – Информатика и вычислительная техника

Москва - 2011.

**Этапы разработки агентной модели «Реализация прохладительных напитков (на примере рынка яблочного сока) »
в имитационной среде Any Logic**

Задача моделирования заключается в том, что конкурирующие компании- производители на рынке яблочного сока «Лебединский», «Мултон», «ВБД» выступают в качестве агентов- продавцов, агенты-покупатели – это потребители яблочного сока.

Имитационная модель по данной теме состоит из трех компонентов: главное меню с параметрами симуляции, класс состояния агентов, моделируемое пространство.

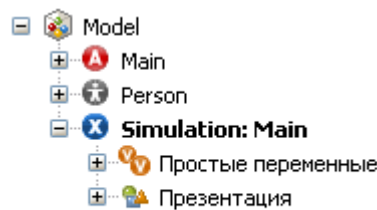


Рис. 18. Параметры симуляции, класс состояния агентов

Главное меню

Рынок сока

Страница настроек эксперимента

Запустить модель и открыть презентацию класса Main

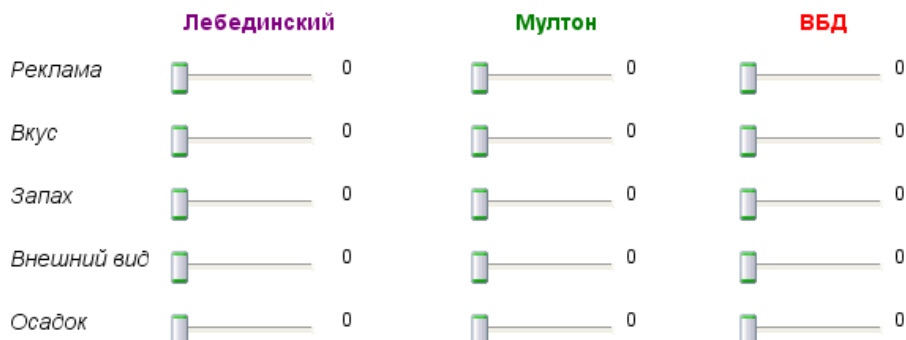


Рис. 19. Страница настроек эксперимента

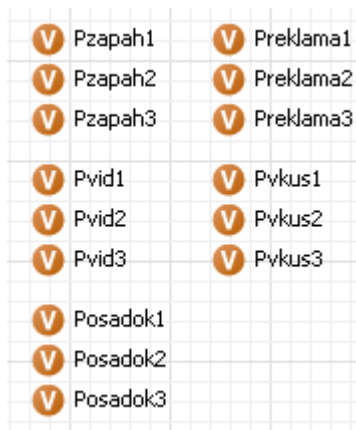


Рис. 20. Переменные состояния

На главном меню располагается несколько компонентов:

- Кнопка запуска модели (генерируется автоматически)
- 15 объектов типа slider, которые позволяют задать параметры покупаемого продукта. Данный элемент располагается на вкладке «Элементы управления» в Any Logic. Для того чтобы объект обладал поведением ему необходимо задать начальные параметры.

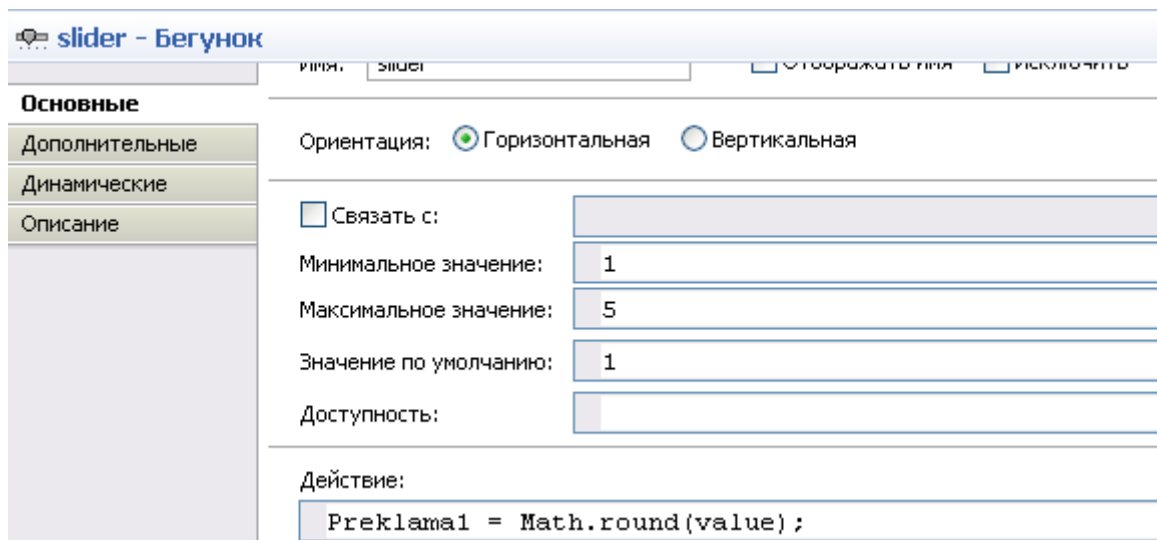


Рис. 21. Экранная форма оформления действий slidera – бегунка

Устанавливаем минимальные и максимальные значения, а так же вписываем действие, которые необходимо выполнить при перемещении ползунка. Данный код означает, что в переменную Preklama1 записывается значение текущего состояния элемента.

Такие действия прописываем для каждого slidera. Переменные находятся на этом же экране.

Экран активного агента

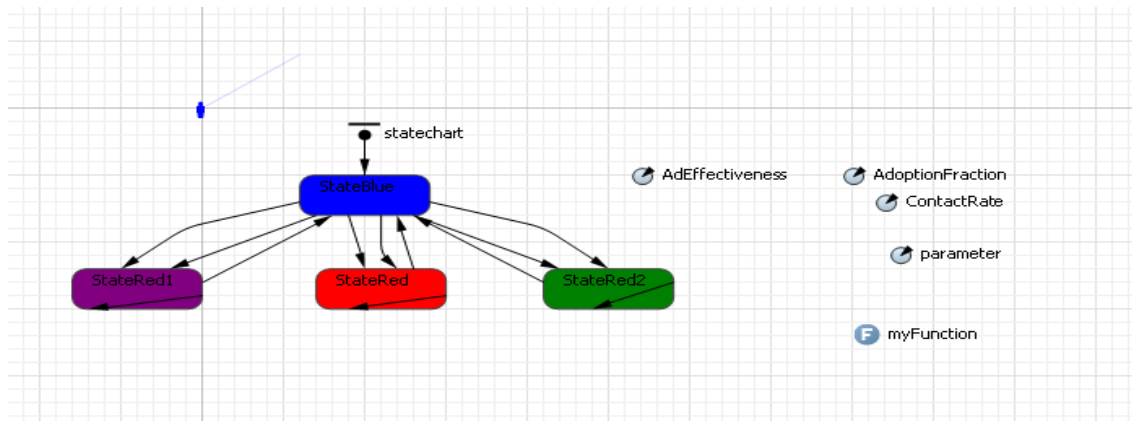


Рис. 22. Экран активного агента

Для начала создадим еще 15 параметров, таких же, как и на главном меню. Это делается для передачи начальных значений данных каждому агенту.

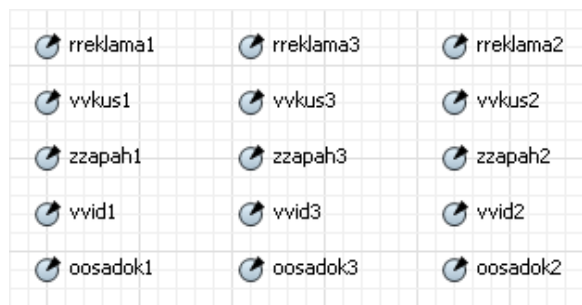
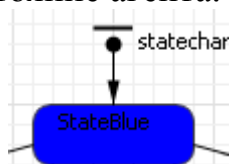


Рис. 23. Параметры состояния

Теперь добавим еще несколько переменных:

- AdEffectiveness – это эффективность, с которой агенты взаимодействуют друг с другом. Устанавливаем по умолчанию значение 0.011;
- AdoptionFraction – внушаемость агента;
- ContactRate – максимальное количество человек, с которым может взаимодействовать агент;
- Parameter – вероятность, что наш агент по сообщению согласится купить новый сок

Состояние агента:



1. Это начальное состояние агента, означающее, что он еще не купил продукт. Для того чтобы в модели агент окрасился таким же синим цветом пропишем его код `person.setFillColor (blue);`

2.

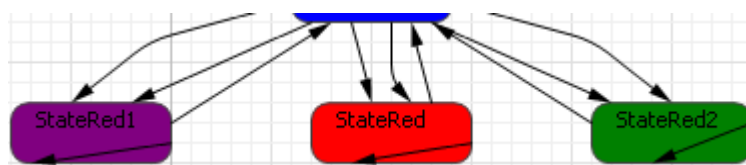


Рис. 24. Отображение состояние агентов


На экранной форме отображены три состояния агента, означающие что агент купил один из трех предложенных продуктов. В коде прописываем изменение цвета:

```
person.setFillColor( purple );  
person.setFillColor( red );  
person.setFillColor( green );
```

От начального состояния к этим трем состояниям создаем переходы (3 стрелочки): первая стрелочка означает, что агент переходит из состояния «без покупки» к состоянию «покупки». Интенсивность означает, как часто агент совершит покупку. Для этого вписываем формулы, зависящие от заданных на первом экране параметров. Так же допишем в дополнительное условие код *randomTrue(parameter)*, которое будет отвечать за вероятность покупки.

| | | | | | |
|----------------------------|--|---|------------------------------------|--|--|
| Имя: | <input type="text" value="transition4"/> | <input type="checkbox"/> Отображать имя | <input type="checkbox"/> Исключить | <input type="checkbox"/> На верхнем уровне | <input checked="" type="checkbox"/> На презентации |
| Происходит: | С заданной интенсивностью <input type="button" value="v"/> | | | | |
| С заданной интенсивностью: | <input type="text" value="(AdEffectiveness*(rreklamal+vvkusl+zzapah1+vvidl+oosadok1)/5)"/> | | | | |
| Действие: | <input type="text"/> | | | | |
| Доп. условие: | <input type="text" value="randomTrue(parameter)"/> | | | | |

Рис. 25. Экранная форма для задания интенсивности

Далее создадим внутреннюю стрелочку . Она будет посылать сообщения «родственникам» агента о том, чтобы они приобрели товар, которые в свою очередь, будут посылать сообщение зависимым агентам.

| | | | |
|----------------------------|---|---|------------------------------------|
| Имя: | <input type="text" value="transition8"/> | <input type="checkbox"/> Отображать имя | <input type="checkbox"/> Исключить |
| Происходит: | <input type="button" value="С заданной интенсивностью"/> | | |
| С заданной интенсивностью: | <input type="text" value="ContactRate*AdoptionFraction"/> | | |
| Действие: | <input type="text" value='send("Buy1!", RANDOM_CONNECTED);'/> | | |
| Доп. условие: | <input type="text"/> | | |

Рис. 26. Экранная форма задания интенсивности и сообщения агентам

Здесь же задается интенсивность сообщения. Прописываем, какое сообщение будем отправлять `send("Buy1!", RANDOM_CONNECTED);`. `.RANDOM_CONNECTED` означает, что сообщение будет послано случайному агенту из «родственников».

Для того, чтобы агент смог прочитать сообщение создадим третью стрелочку и пропишем в ней следующее:

| | |
|-----------------------|---|
| Происходит: | <input type="button" value="При получении сообщения"/> |
| Тип сообщения: | <input type="radio"/> boolean <input type="radio"/> int <input type="radio"/> double <input checked="" type="radio"/> String <input type="radio"/> Другой |
| Имя класса: | <input type="text"/> |
| Осуществлять переход: | <input type="radio"/> Безусловно <input checked="" type="radio"/> Если сообщение равно <input type="text" value="Buy1!"/> <input type="radio"/> Если выполняется условие |
| Действие: | <input type="text"/> |
| Доп. условие: | <input type="text" value="randomTrue (AdoptionFraction)"/> |

Рис. 27. Создание сообщения

И, наконец, смоделируем повторные покупки. Для этого создадим четвертый переход (в обратную сторону) и впишем `get_Main()` – ссылка на главный класс.

| | |
|--------------|--|
| Происходит: | <input type="button" value="По таймауту"/> |
| По таймауту: | <input type="text" value="get_Main().Time"/> |

Рис.28. Создание перехода (в обратную сторону)

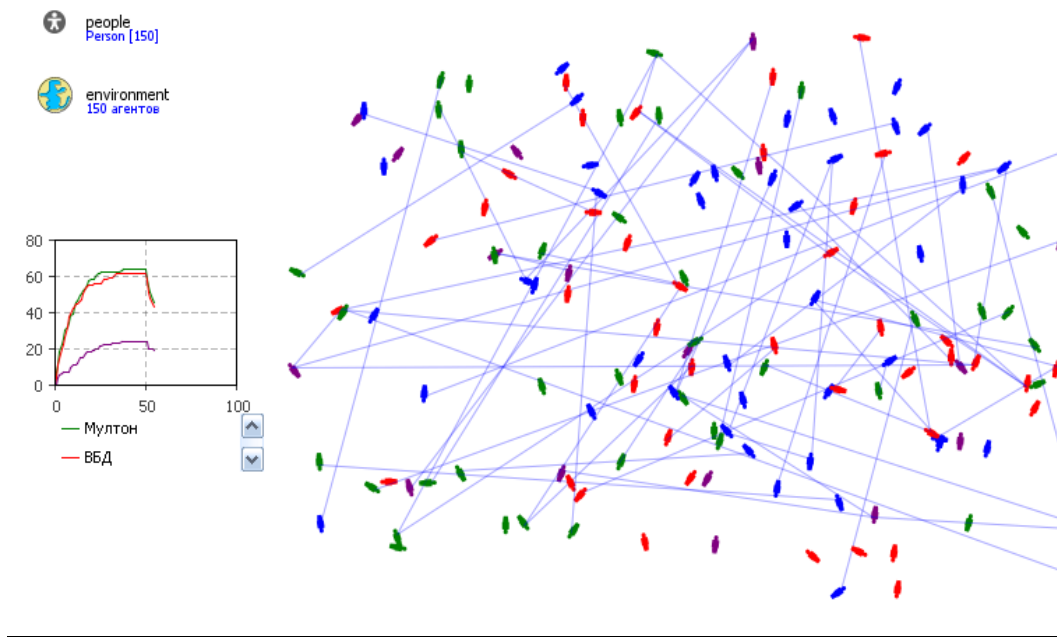


Рис. 29. Экранная форма агентной модели

На экранной форме представлены основные два элемента, это среда и агент.

Среда:

| | | | | |
|-------------------------|--|-------------------------------------|------------------------------------|---------------------------|
| Тип пространства: | <input checked="" type="radio"/> Непрерывное2D | <input type="radio"/> Непрерывное3D | <input type="radio"/> Дискретное2D | <input type="radio"/> ГИС |
| Ширина: | 650 | | | |
| Высота: | 400 | | | |
| Z-Высота: | 0 | | | |
| Столбцы: | 100 | | | |
| Строки: | 100 | | | |
| Тип соседства: | Мурово | | | |
| Тип расположения: | Случайное | <input checked="" type="checkbox"/> | Применить при запуске | |
| Тип сети: | Согласно расстоянию | <input checked="" type="checkbox"/> | Применить при запуске | |
| Кол-во связей у агента: | 2 | | | |
| Радиус соединения: | 20 | | | |
| Доля соседей: | 0.95 | | | |
| M: | 10 | | | |

Рис. 30. Экранная форма настроек моделирования

Настройки означают, что на начало моделирования будет создано пространство размером 650 на 400. Агенты взаимодействуют в зависимости от расстояния, в данном случае 20.

Агент:

Делаем соединение начальных и модельных параметров

| | |
|-----------------|--|
| Основные | Тип: <input type="text" value="Person"/> |
| Параметры | Пакет: <input type="text" value="model"/> |
| Статистика | Среда: <input type="text" value="environment"/> |
| Описание | |
| | AdEffectiveness: <input type="text" value="0.011"/> |
| | ContactRate: <input type="text" value="100"/> |
| | AdoptionFraction: <input type="text" value="0.015"/> |
| | rreklama1: <input type="text" value="reklama1"/> |
| | rreklama2: <input type="text" value="reklama2"/> |
| | rreklama3: <input type="text" value="reklama3"/> |
| | vvkus1: <input type="text" value="vkus2"/> |
| | vvkus2: <input type="text" value="vkus2"/> |
| | vvkus3: <input type="text" value="vkus3"/> |

Рис. 31. Экранная форма по соединению начальных и модельных параметров

Моделируем рисунок для анимации агента



Рис. 32. Рисунок анимации модели

Если в модели предпочтительнее движение агентов, прописываем действия при запуске код

```
moveTo( uniform( getEnvironment().getWidth() ), uniform
( getEnvironment().getHeight() ) );
```

для графической интерпретации задаем параметры:

| | | |
|--|--|---|
| <input checked="" type="radio"/> Значение | <input type="radio"/> Набор данных | Заголовок: <input type="text" value="Мултон"/> |
| Значение: <input type="text" value="people.NGreen()"/> | | |
| Стиль маркера: | <input type="text" value="—"/> | Цвет: <input type="text" value="green"/> |
| <input checked="" type="checkbox"/> Рисовать линию | Толщина линии: <input type="text" value="1 pt"/> | Интерполяция: <input type="text" value="Линейная"/> |
| <input checked="" type="radio"/> Значение | <input type="radio"/> Набор данных | Заголовок: <input type="text" value="ВБД"/> |
| Значение: <input type="text" value="people.NRed()"/> | | |
| Стиль маркера: | <input type="text" value="—"/> | Цвет: <input type="text" value="red"/> |
| <input checked="" type="checkbox"/> Рисовать линию | Толщина линии: <input type="text" value="1 pt"/> | Интерполяция: <input type="text" value="Линейная"/> |
| <input checked="" type="radio"/> Значение | <input type="radio"/> Набор данных | Заголовок: <input type="text" value="Лебединский"/> |
| Значение: <input type="text" value="people.NSir()"/> | | |
| Стиль маркера: | <input type="text" value="—"/> | Цвет: <input type="text" value="purple"/> |
| <input checked="" type="checkbox"/> Рисовать линию | Толщина линии: <input type="text" value="1 pt"/> | Интерполяция: <input type="text" value="Линейная"/> |

Рис. 33. Экранная форма для задания параметров графической интерпретации модели

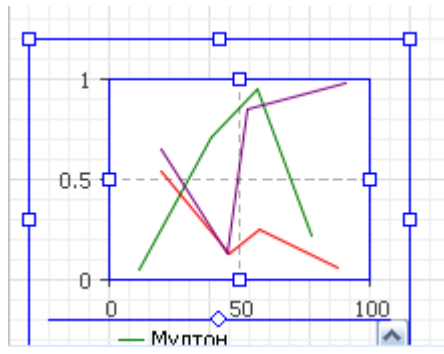


Рис. 34. Графическая интерпретация модели