

*Макарова Наталья Владимировна,  
Сениченков Юрий Борисович,  
Титова Юлияна Францевна*

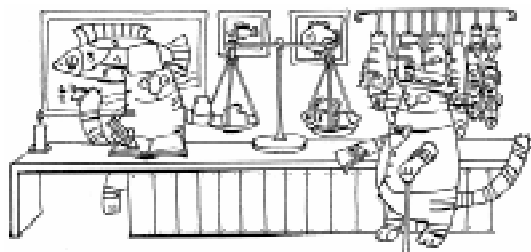
## ШКОЛА МОДЕЛИРОВАНИЯ 2003.

### ЗАНЯТИЕ 5.

#### МОДЕЛИРОВАНИЕ СИСТЕМ.

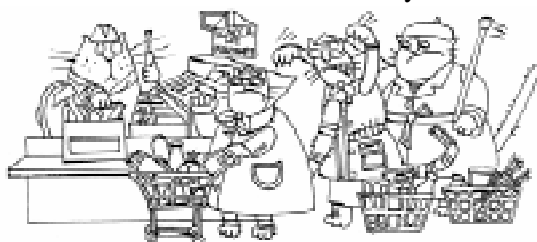
#### МАЛЕНЬКАЯ МОДЕЛЬ БОЛЬШОГО УНИВЕРСАМА

Современные большие магазины породили большие проблемы, разрешить которые совсем не просто. Каких-нибудь сто лет назад все оборудование магазина составляли прилавок, кладовка, весы, да книга с записями о сделанных закупках и продажах.



Если в магазине были хозяин, помощник и рассыльный, то можно было говорить о большом «деле». Можно ли сравнить их с сегодняшним магазинным хозяйством?! Склады современных магазинов оборудованы компьютерами со специальными базами данных, позволяющими за доли секунды узнать, сколько чего закуплено, почем, какого качества, и как данный товар раскупается. Компьютеры помогают кассирам, товароведом, службе охраны. Внутри магазина и снаружи стоят телевизионные камеры – целая телестудия. Специальные автоматические системы поддерживают нужные температурные режимы на складах и в залах. А сколько это все стоит! Естественно, что хозяева магазина хотят, чтобы вложенные в магазин деньги приносили прибыль. А это означает, что оборудование и обслуживающий персонал должны работать эффективно. Как узнать, сколько касс должно быть в магазине, чтобы с одной стороны они не

простаивали, а с другой стороны, чтобы из-за их нехватки не создавалось бы очереди? Как узнать, сколько товара нужно закупать, чтобы он не залеживался, и всегда был на прилавке свежим? Готовые формулы в данном случае найти трудно, но можно попытаться построить компьютерную модель и на ней «разыгрывать» различные сценарии. Например, предположим, что в час пик сломалась одна касса. Насколько увеличится



время обслуживания одного покупателя? Или перед праздником возник наплыв посетителей – сколько дополнительных рабочих необходимо привлечь? Моделируя поведение отдельного покупателя, работу касс и других магазинных служб, можно попытаться ответить на такие вопросы. Модели, в которых задаются законы поведения отдельного элемента и правила его взаимодействия с окружением, а изучается поведение всей системы, называются имитационными. Попробуем построить имитационную модель магазина, понимая, что она, конечно же, чрезвычайно проста, но при желании может быть превращена в практически важную, приносящую реальную пользу модель.

Предположим, что мы находимся в универсаме днем, когда работает одна кас-

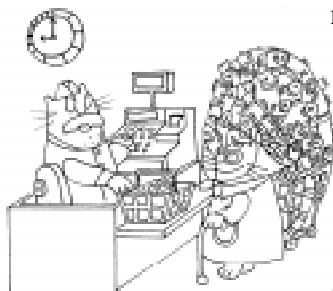
са, обслуживающая в основном старушек, пришедших за повседневным товаром – молоком, свежим хлебом, сыром и маслом.



Такие покупки редко делают впрок, и мы можем приблизительно оценить максимальное и минимальное количество покупаемых продуктов, исходя из дневной потребности. Можно считать, что количество одного вида товара, необходимое «статистической» старушке, равномерно распределено между заданным максимальным и минимальным значениями. Простоты ради можно считать, что в магазине только один сорт молока, хлеба, масла и сыра, или, что то же самое, что мы работаем со средними ценами.

Процесс обслуживания чрезвычайно прост – старушка стоит в очереди и подходит к кассе, услышав привычное «следующий!», а затем выкладывает товар перед кассиром. Выложив товар, она произносит: «Милая, (сосчитай)», и с этого момента начинается подсчет итоговой суммы, пробиваемой на чеке. Время предъявления товара и время подсчета носит также случайный характер и распределено по заданному случайному закону, как и количество товара в корзинке.

Нас может интересовать среднее время обслуживания одной старушки в зависимости от числа касс и скоростей работы кассиров. Или средняя сумма приобретенных товаров одной старушкой в зависимости от вида законов, характеризующих дневные потребности – в будние дни покупки одни, перед праздниками – другие. С математической точки зрения данная зада-



ча решается очень просто: надо случайным образом «создать» сто корзинок с товарами – ста векторам, состоящим из четырех компонентов, присвоить заданные значения. Далее следует случайным образом приписать каждой старушке время предъявления товара и время подсчета итоговой суммы. Осталось усреднить сумму времени предъявления товара и времени подсчета, чтобы получить среднее время обслуживания и просуммировать все чеки. Кажется бы, что пакет Model Vision для этих целей использовать не нужно. Однако взглянем на эту задачу несколько по-другому. Предположим, что мы хотим как можно точнее сохранить структуру исходной задачи в нашей компьютерной модели: нам нужен и универсам, в котором есть касса, возможно не одна, и покупатели – бабушки. Иными словами, мы хотим составлять сложную модель из отдельных кубиков. Такую модель легко модифицировать, меняя один кубик на другой, если в том появилась необходимость. Мы хотим также, чтобы покупатели и кассиры общались друг с другом как в обычной жизни, передавая друг другу сообщения. Это позволит нам лучше понимать логику взаимодействия между компонентами. В этом случае следует использовать Model Vision.

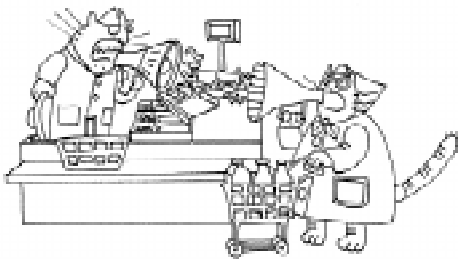
Пропустим этап предварительного, более строгого, описания нашей модели, когда мы назначаем имена переменных, выбираем законы распределения, конкретизируем все зависимости, которые мы хотим изучить – в данном случае все это интуитивно понятно, и опишем сразу компьютерную модель.

Модель, реализованная в Model Vision Studium, состоит из трех компонентов: сложного, или составного компонента – «Универсам», и двух простых, элементарных – «Бабушка» и «Касса» (рисунок 1).

Как видно из рисунка, графический образ составного компонента отличается от образа элементарного: составной имеет двойную рамку. Если мы хотим узнать, что скрывается за изображением составного компонента, достаточно установить курсор в поле прямоугольника и щелкнуть мышью. Откроется окно, показывающее его внутрен-

ную структуру. Теперь понятно, что в нашей модели (проект с условным названием «Бабушка») существует три класса – «Универсам», «Бабушка» и «Касса» (рисунок 2).

Последовательность создания классов такова. Сначала создаем класс «Бабушка» (рисунок 3), затем – класс «Касса» (рисунок 4). Мы видим, что оба класса обладают входами и выходами, на которые подаются специальные переменные, называемые сигналами. Они носят имена «Милая» и «Следующий». Смысл их понятен –



кассирша приглашает очередного покупателя привычным возгласом «Следующий», и начинает считать, когда очередная старушка произносит «Милая (сосчитай)». Если время возникновения сигналов можно формировать случайным образом, то можно управлять и потоком старушек, и расторопностью кассиров. Переменная «товары» нужна для того, чтобы кассирша могла сформировать переменную «чек» для очередной старушки, а переменная «Итого» – чтобы зафиксировать общую сумму покупок. Для того, чтобы смоделировать очередь из ста

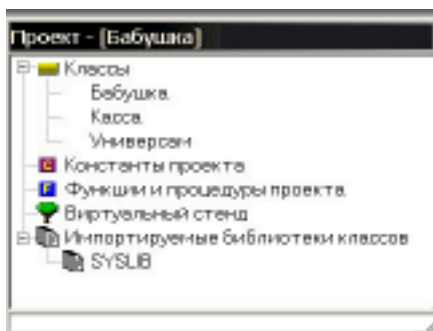


Рисунок 2. Классы проекта «Бабушка».

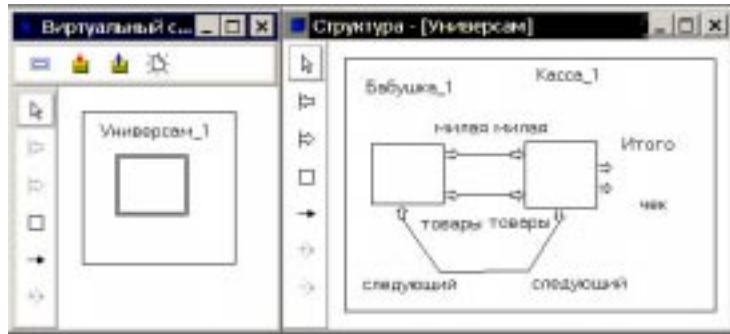


Рисунок 1. Составной объект «Универсам», состоит из элементов, не имеющих внутренней структуры – «Бабушка\_1» и «Касса\_1».

старушек, мы применили простой прием – в классе «Бабушка» при получении сигнала «следующий» мгновенно меняется содержимое корзины, и перед кассиром предстает «новая» старушка. Как в театре одного актера, играющего сразу несколько персонажей и передевающегося на глазах зрителей.

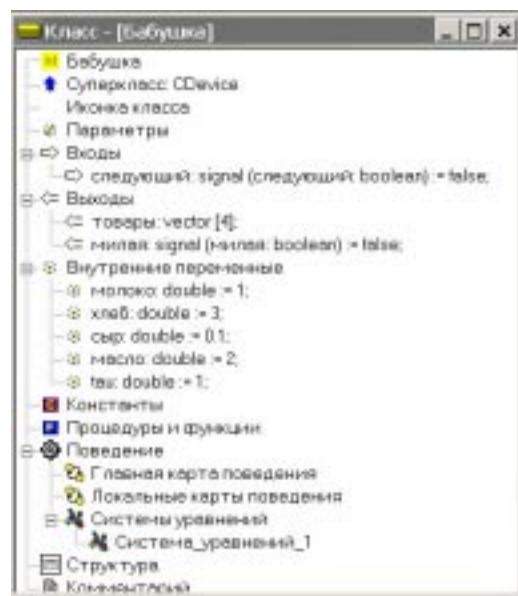


Рисунок 3. Класс «Бабушка».

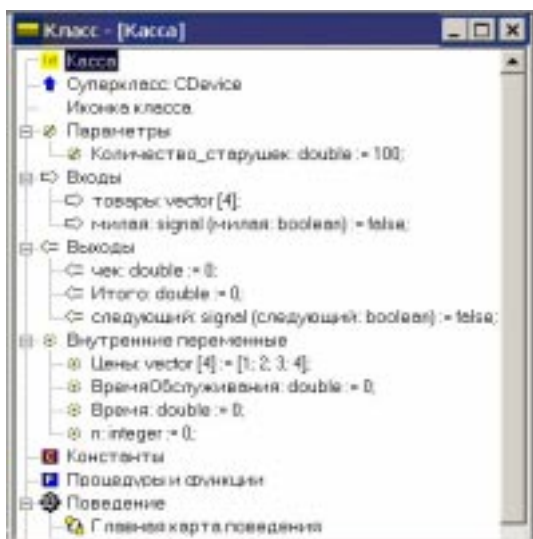


Рисунок 4. Класс «Касса».

Класс «Универсам» не содержит собственных переменных. Важна только его «Структура». Создайте новый класс и дайте ему имя «Универсам». Установите курсор в поле «Структура» и щелкните мышью.

Откроется окно с тем же названием. Перенесите в него мышью экземпляр класса «Бабушка» – появится объект «Бабушка\_1» – и экземпляр класса «Касса» – «Касса\_1». Можно сразу же соединить соответствующие «сигнальные» входы и выходы и вход и выход с именем «товар», как показано на рисунке 1.

Посмотрим, как происходит взаимодействие между экземплярами классов «Бабушка» и «Касса». Очередная бабушка подходит к кассе, только услышав возглас «Сле-

дующий». Одноименная переменная – это входная переменная класса «Бабушка», имеющая тип «сигнал». В корзине старушки (вектор «товары») уже лежат хлеб, молоко, сыр и масло. Она начинает выкладывать товары, и, закончив, сообщает об этом кассиру «Милая (сосчитай, пожалуйста!)». После чего ждет, когда можно будет расплатиться и пойти домой. Работа кассира начинается и завершается фразой «Следующий». Посмотрите, как эта последовательность действий отражена на карте поведения класса «Бабушка» (рисунок 5). Мы видим три состояния – «Init» (название этого состояния невозможно изменить), «Выкладываем» и «Ждем». В первом состоянии «Init» бабушка стоит в очереди и ждет, когда кассирша освободится, затем она выкладывает товар – состояние «Выкладываем», и, наконец, ждет чек, расплачивается и уходит, что совпадает по времени с пришедшим от кассира сигналом новой старушки – «Ждем». Обратите внимание на команду «Send». Это и есть команда, с помощью которой можно переслать сигнал другому компоненту. В данном случае посылает сигнал компонент «Бабушка», а принимает его компонент «Касса». Теперь посмотрите на карту поведения компонента «Касса» (рисунок 6). Кассирша посылает сигнал «Следующий» и ждет, когда придет от старушки сигнал «Милая», после чего подсчитывает результат.

Теперь поговорим о расчетах. Входные действия узла «Init» класса «Бабушка» «наполняют» корзину очередной старушки. Переменная, названная по имени товара, умножается на случайное число – «uniform (0, 1)», распределенное по равномерному закону от минимального (нуля) до максимального значения (единица). Так же случайно формируется и время предъявления товара – «tau». Эту переменную Вы видите на карте поведения класса «Бабушка» рядом со стрелкой из состояния «Выкладываем» в состояние «Ждем». Она появляется там в сочетании со словом «after». Это означает, что переход из состояния «Выкладываем» в состояние «Ждем» автоматически произойдет через «tau» единиц модельного времени. В классе «Кассир» мгновенные действия связаны только с формированием переменной «Чек»

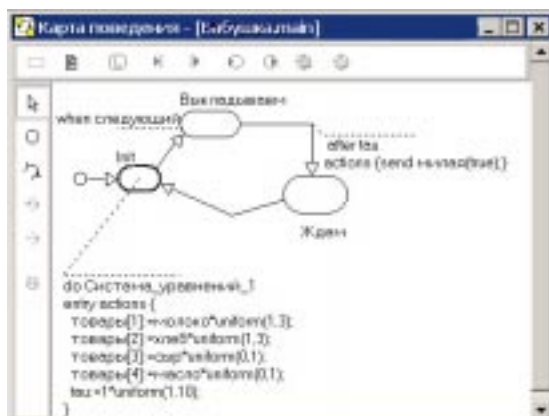
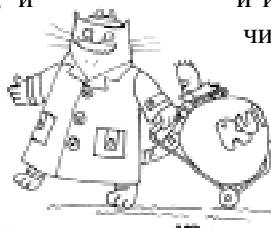
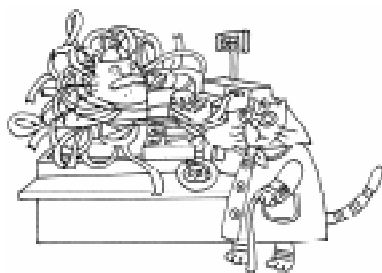


Рисунок 5. Карта поведения класса «Бабушка».

и «Итого». Осталось разобраться со средним временем обслуживания.

В нашей модели кассирша работает непрерывно – подает команду «Следующий», ждет, пока выложат товар, и подсчитывает сумму для очередной старушки. Следовательно, общее время работы модели, поделенное на число обслуженных покупателей, и есть среднее время (переменная «Время»), потраченное кассиршей на одну старушку. Еще одна глобальная характеристика – выручка магазина – «Итого».



Визуализация модели может быть очень простой. Во-первых, можно нарисовать график выручки от времени. Во-вторых, можно показать «характеристики» процесса обслуживания: число обслуженных покупателей, время подсчета цены товаров, среднее время обслуживания старушки (рисунок 7).

Заканчивая описание этой простейшей модели, хочется еще раз подчеркнуть – эта простая модель существенно отличается от рассмотренных на предыдущих занятиях. Здесь мы описали только

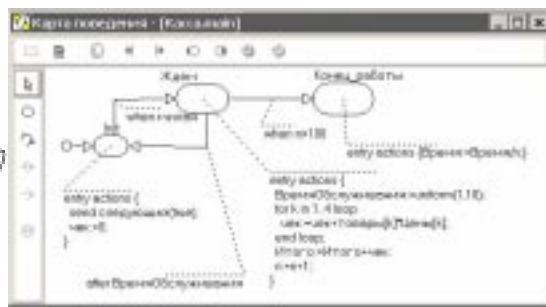


Рисунок 6. Карта поведения класса «Касса».

законы поведения отдельных компонентов и их взаимодействия друг с другом, а получили в результате закономерности поведения всей системы. Составление этой модели скорее напоминает игру, в которой мы определяем или меняем отдельные правила, и смотрим, как это отразится на поведении в целом. Если она увлекла Вас, попробуйте поместить в модель универсама две кассы, измените законы распределения товаров или времени обслуживания.

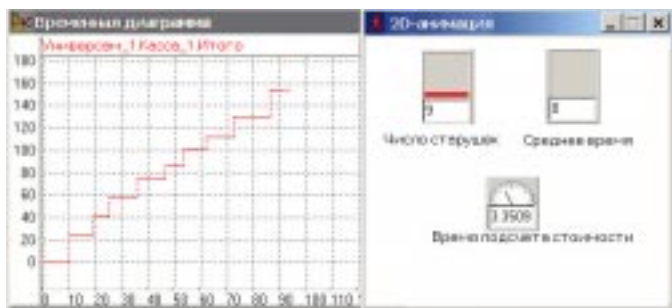


Рисунок 7. Вычислительный эксперимент.

**Макарова Наталья Владимировна,**  
доктор пед. наук, канд. техн. наук,  
проректор МБИ (СПб),  
зав. каф. информационных систем  
и технологий,

**Сениченков Юрий Борисович,**  
доцент кафедры Распределенных  
вычислений и компьютерных сетей  
Санкт-Петербургского  
Политехнического Университета,

**Титова Юлия Францевна,**  
канд. пед. наук, доцент кафедры  
информационных систем и  
технологий Международного  
Банковского Института.

