МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Методические указания к практическим работам по дисциплине “Математическое моделирование”

Уровень основной образовательной программы: **магистратура**
Направление подготовки магистра: **09.04.04 «Бизнес-информатика»**
Форма обучения: **очная**
Факультет систем управления (ФСУ)
Кафедра автоматизации обработки информации (АОИ)
Курс 1 Семестр 1, 2

Разработчики:
профессор каф. АОИ
_____Н.В. Замятин

Томск 2018

СОДЕРЖАНИЕ

Введение……..……..2
1. Практическая работа 1. Исследование предметной области моделирования……. 2
2. Практическая работа 2. Пакет SIMULINK и визуальное моделирование ……5
3. Практическая работа 3. Математические модели на основе обыкновенных дифференциальных уравнений (ОДУ) ……....22
4. Практическая работы 4. Методология структурного синтеза моделей …….……..…….……28
5. Практическая работы 5. Пакет AnyLogic для моделирования сложных систем…….. ……..……..38
6. Библиографический список……47

1. ВВЕДЕНИЕ

Цель изучения дисциплины

Задачей математического моделирования как научного направления является воссоздание с помощью компьютера представление и исследование моделей различных предметных областей. Из всего многообразия научных и технических исследований, определяемых математическим моделированием, в учебной дисциплине «Математическое моделирование» выбраны направления, связанные с проблемами аналитического, имитационного моделирования а также принципами визуального моделирования и агентного моделирования, а также моделирования бизнес-процессов.

Цель изучения дисциплины - ознакомление студентов с методами математического моделирования, представления и обработки данных и знаний для построения моделей.

1. Практическое занятие 1. Исследование предметной области (8 часов)

Цель занятия. Изучить заданную предметную область и построить модель в виде графа.

Методические указания. Для построения модели в виде графа необходимо выполнить следующие шаги:

1) Определить целевые действия задачи (являющиеся решениями).

2) Определить промежуточные действия или цепочку действий, между начальным состоянием и конечным (между тем, что имеется, и целевым действием).

3) Опередить условия для каждого действия, при котором его целесообразно и возможно выполнить. Определить порядок выполнения действий.

4) Добавить конкретные факты , исходя из поставленной задачи.

5) Преобразовать полученный порядок действий и соответствующие им факты, условия и действия.

6) Для проверки правильности построения записать цепочки, явно проследив связи между ними. Этот набор шагов предполагает движение при построении модели от результата к начальному состоянию, но возможно и движение от начального состояния к результату (шаги 1 и 2).

7) Присвоить обозначения фактам Ф, правилам П, действиям Д.

8) Построить граф предметной области. (пример рис.1)

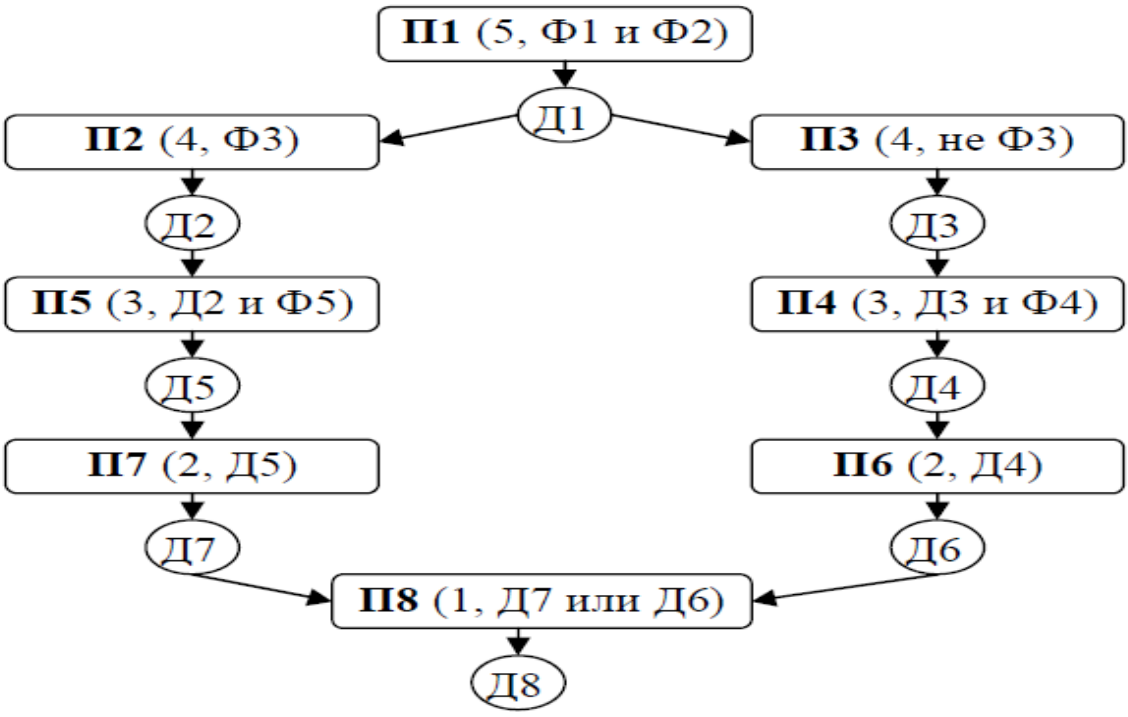


Рис. 1 – Пример графа модели знаний

Варианты заданий

1. Построить модель представления знаний в предметной области «Железная дорога» (продажа билетов).
2. Построить модель представления знаний в предметной области «Торговый центр» (организация).
3. Построить модель представления знаний в предметной области «Автозаправка» (обслуживание клиентов).
4. Построить модель представления знаний в предметной области «Компьютерные сети» (организация).
5. Построить модель представления знаний в предметной области «Университет» (учебный процесс).
6. Построить модель представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).
7. Построить модель представления знаний в предметной области «Компьютерная безопасность» (угрозы).
8. Построить модель представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).
9. Построить модель представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).
10. Построить модель представления знаний в предметной области «Туристическое агентство» (работа с клиентами).
11. Построить модель представления знаний в предметной области «Кухня» (приготовление пищи).
12. Построить модель представления знаний в предметной области «Больница» (прием больных).
13. Построить модель представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).
14. Построить модель представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).

15. Построить модель представления знаний в предметной области «Операционные системы» (функционирование).

16. Построить модель представления знаний в предметной области «Информационные системы» (виды и функционирование).

17. Построить модель представления знаний в предметной области «Предприятие» (структура и функционирование).

2. Практическая работа №2. ПАКЕТ SIMULINK И ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ (8 часов)

Цель работы изучение пакета Simulink и визуального моделирования.

1 Общие сведения

MATLAB - это высокоэффективный язык инженерных и научных вычислений. Он поддерживает математические вычисления, визуализацию научной графики и программирование с использованием легко осваиваемого операционного окружения, когда задачи и их решения могут быть представлены в нотации, близкой к математической.

Система MATLAB - это одновременно и операционная среда и язык программирования. Одна из наиболее сильных сторон системы состоит в том, что на языке MATLAB могут быть написаны программы для многократного использования. Пользователь может сам написать специализированные функции и программы, которые оформляются в виде М-файлов. По мере увеличения количества созданных программ возникают проблемы их классификации и тогда можно попытаться собрать родственные функции в специальные папки. Это приводит к концепции пакетов прикладных программ, которые представляют собой коллекции М-файлов для решения определенной задачи или проблемы. Кроме того, широкий выбор различных пакетов расширения системы MATLAB позволяет быстро выполнить практически любую инженерную задачу и наглядно представить ее решение с помощью имитационной модели. MATLAB — одна из старейших, тщательно

проработанных и проверенных временем систем автоматизации математических расчетов, построенная на расширенном представлении и применении матричных операций. Это нашло отражение в названии системы — MATrix LABoratory — матричная лаборатория. Однако синтаксис языка программирования системы продуман настолько тщательно, что эта ориентация почти не ощущается теми пользователями, которых не интересуют непосредственно матричные вычисления. Примером служит расширение MATLAB — Simulink, которая далеко вышла за пределы специализированной матричной системы и стала одной из наиболее мощных универсальных интегрированных сред. Слово «интегрированная» указывает на то, что в этой системе объединены удобная оболочка, редактор выражений и текстовых комментариев, вычислитель и графический программный процессор. В новой версии используются такие мощные типы данных, как многомерные массивы, массивы ячеек, массивы структур, массивы Java и разреженные матрицы, что открывает возможности применения системы при создании и отладке новых алгоритмов матричных и основанных на них параллельных вычислений и крупных баз данных.

Система MATLAB разработана Молером (C. B. Moler) и с конца 70-х гг. широко использовалась на больших ЭВМ. В начале 80-х гг. Джон Литл (John Little) из фирмы MathWorks, Inc. разработал версии системы PC MATLAB для компьютеров класса IBM PC, VAX и Macintosh. В дальнейшем были созданы версии для рабочих станций Sun, компьютеров с операционной системой UNIX и многих других типов больших и малых ЭВМ. Сейчас свыше десятка популярных компьютерных платформ могут работать с системой MATLAB. К расширению системы были привлечены крупнейшие научные школы мира в области математики, программирования и естествознания. Матлаб непрерывно развивается и теперь появилась новейшая версия этой системы — MATLAB 7. Одной из основных задач системы было предоставление пользователям мощного языка программирования, ориентированного на

математические расчеты и способного превзойти возможности традиционных языков программирования, которые многие годы использовались для реализации численных методов. При этом особое внимание уделялось как повышению скорости вычислений, так и адаптации системы к решению самых разнообразных задач пользователей.

Возможности MATLAB весьма обширны, а по скорости выполнения задач система нередко превосходит своих конкурентов. Она применима для расчетов практически в любой области науки и техники. Например, широко используется при математическом моделировании механических устройств и систем, в частности в динамике, гидродинамике, аэродинамике, акустике, энергетике и т. д. Этому способствует не только расширенный набор матричных и иных операций и функций, но и наличие пакета расширения (toolbox) Simulink, специально предназначенного для решения задач блочного моделирования динамических систем и устройств, а также десятков других пакетов расширений.

В системе MATLAB содержатся специальные средства для электротехнических и радиотехнических расчетов (операции с комплексными числами, матрицами, векторами и полиномами, обработка данных, анализ сигналов и цифровая фильтрация), обработки изображений, реализации нейронных сетей, а также средства, относящиеся к другим новым направлениям науки и техники. Они иллюстрируются множеством практически полезных примеров.

Важными достоинствами системы являются ее открытость и расширяемость. Большинство команд и функций системы реализованы в виде текстовых m-файлов (с расширением m) и файлов на языке Си, причем все файлы доступны для модификации. Пользователю дана возможность создавать не только отдельные файлы, но и библиотеки файлов для реализации специфических задач.

Легкость модификации системы и возможность ее адаптации к решению задач науки и техники привели к созданию десятков пакетов прикладных программ (toolbox), расширивших сферы применения системы. Некоторые из них, например Notebook (интеграция с текстовым процессором Word и подготовка «живых» электронных книг), Symbolic Math и Extended Symbolic Math (символьные вычисления с применением ядра системы Maple V R5) и Simulink (моделирование динамических систем и устройств, заданных в виде системы блоков), настолько органично интегрировались с системой MATLAB, что стали ее составными частями.

В последние годы разработчики математических систем уделяют огромное внимание их интеграции и совместному использованию. Это не только расширяет класс решаемых каждой системой задач, но и позволяет подобрать для них самые лучшие и наиболее подходящие инструментальные средства. Решение сложных математических задач сразу на нескольких системах существенно повышает вероятность получения корректных результатов — увы, как математики так и математические системы способны ошибаться, особенно при некорректной постановке задач неопытными пользователями.

С системой MATLAB могут интегрироваться такие популярные математические системы, как Mathcad, Maple V и Mathematica.

Новые свойства системе MATLAB придала ее интеграция с программной системой Simulink, созданной для моделирования динамических систем и устройств, заданных в виде системы блоков. Базируясь на принципах визуально-ориентированного программирования, Simulink позволяет выполнять моделирование сложных устройств с высокой степенью достоверности и с средствами представления результатов. В свою очередь, многие другие математические системы, например Mathcad и Maple, допускают установление объектных и динамических связей с системой

MATLAB, что позволяет использовать в них эффективные средства MATLAB для работы с матрицами.

Программа Simulink является приложением к пакету MATLAB. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно общих знаний требующихся при работе на компьютере и знаний той предметной области, в которой он работает.

Simulink является самостоятельным инструментом MATLAB и при работе с ним не требуется знать MATLAB и остальные его приложения. С другой стороны доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink. Часть входящих в состав пакетов имеет инструменты, встраиваемые в Simulink (например, LTI-Viewer приложения Control System Toolbox – пакета для разработки систем управления). Имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств и т.д).

При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

При математическом моделировании пользователь может выбирать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения,

входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Преимущество Simulink заключается также в том, что он позволяет пополнять библиотеки блоков с помощью подпрограмм написанных как на языке MATLAB, так и на языках C + +, Fortran и Ada.

2 Запуск Simulink

Для запуска программы необходимо предварительно запустить пакет MATLAB. Основное окно пакета MATLAB показано на рисунке 1. Там же показана подсказка появляющаяся в окне при наведении указателя мыши на ярлык Simulink в панели инструментов.

Чтобы запустить программу дважды щелкните на иконку 

. Перед Вами откроется рабочая среда, изображенная на рисунке 2.

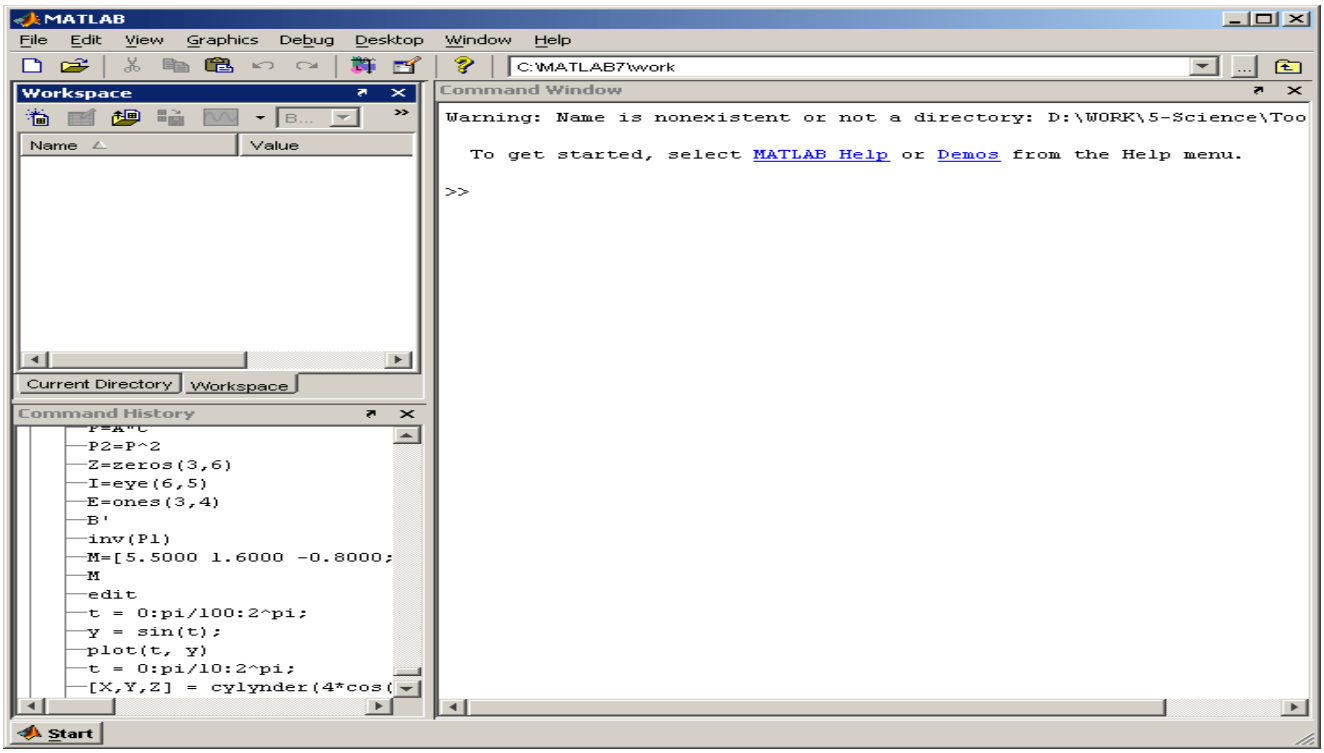


Рисунок 2. Основное окно программы MATLAB

Рабочая среда MatLab 6.x немного отличается от рабочей среды предыдущих версий, она имеет более удобный интерфейс для доступа ко многим вспомогательным элементам

Рабочая среда MatLab содержит следующие элементы:

- панель инструментов с кнопками и раскрывающимся списком;
- окно с вкладками **Launch Pad** и **Workspace**, из которого можно получить доступ к различным модулям ToolBox и к содержимому рабочей среды;
- окно с вкладками **Command History** и **Current Directory**, предназначенное для просмотра и повторного вызова ранее введенных команд, а также для установки текущего каталога;
- командное окно, в котором находится приглашение к вводу » и мигающий вертикальный курсор;
- строку состояния.

Если в рабочей среде MatLab отсутствуют некоторые окна, приведенные на рисунке, то следует в меню **View** выбрать соответствующие пункты: **Command Window**,**Command History**, **Current Directory**, **Workspase**, **Launch Pad**.

Команды следует набирать в командном окне. Символ », обозначающий приглашение к вводу командной строки, набирать не нужно. Для просмотра рабочей области удобно использовать полосы скроллинга или клавиши **Home**, **End**, для перемещения влево или вправо, и **PageUp**, **PageDown** для перемещения вверх или вниз. Если вдруг после перемещения по рабочей области командного окна пропала командная строка с мигающим курсором, просто нажмите **Enter.**

Важно помнить, что набор любой команды или выражения должен заканчиваться нажатием на **Enter**, для того, чтобы программа MatLab выполнила эту команду или вычислила выражение.

После открытия основного окна программы MATLAB нужно запустить программу Simulink. Это можно сделать одним из трех способов:

Нажать кнопку (Simulink) на панели инструментов командного окна MATLAB.

В командной строке главного окна MATLAB напечатать Simulink и нажать клавишу Enter на клавиатуре.

Выполнить команду Open в меню File и открыть файл модели (mdl - файл).

Последний вариант удобно использовать для запуска уже готовой и отлаженной модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель. Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки Simulink (рисунок 3).

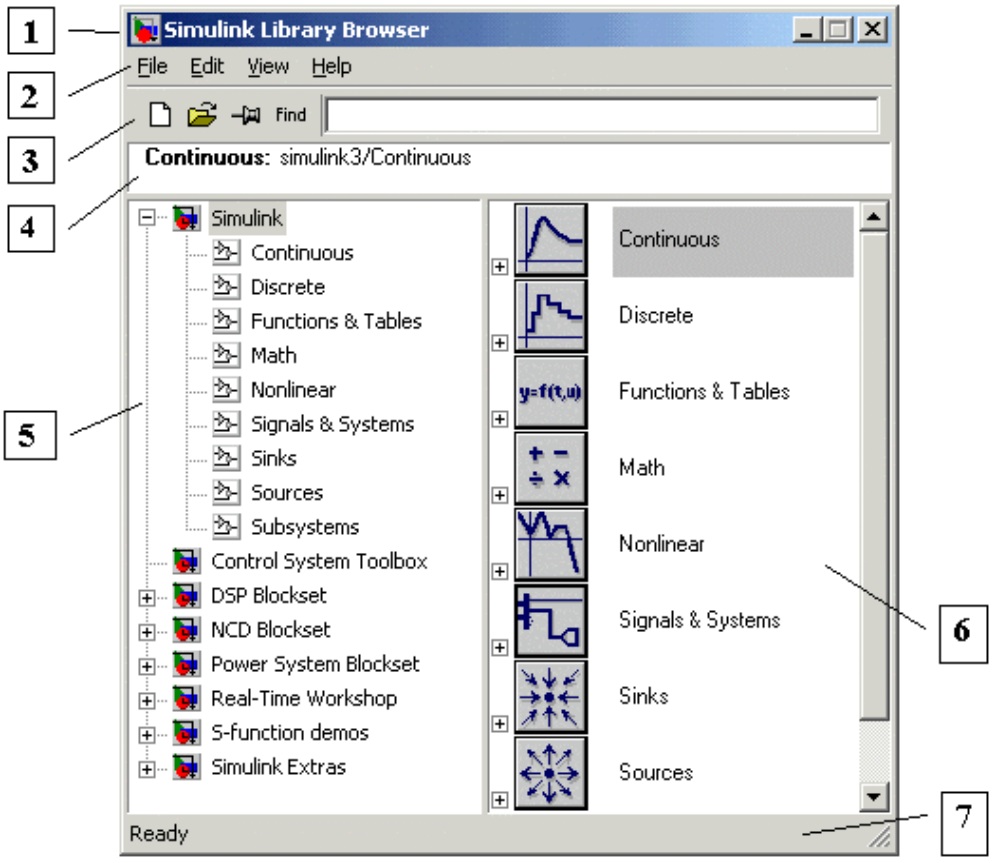


Рисунок 3. Окно обозревателя разделов библиотеки Simulink

3 Обозреватель разделов библиотеки Simulink

Окно обозревателя библиотеки блоков содержит следующие элементы :

1 Заголовок, с названием окна – Simulink Library Browser.

2 Меню, с командами File, Edit, View, Help.

3 Панель инструментов, с ярлыками наиболее часто используемых команд.

4 Окно комментария для вывода поясняющего сообщения о выбранном блоке.

5 Список разделов библиотеки, реализованный в виде дерева.

6 Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков)

7 Строка состояния, содержащая подсказку по выполняемому действию.

На рисунке 2 выделена основная библиотека Simulink (в левой части окна) и показаны ее разделы (в правой части окна).

Библиотека Simulink содержит следующие основные разделы:

Continuous – линейные блоки.

Discrete – дискретные блоки.

Functions & Tables – функции и таблицы.

Math – блоки математических операций.

Nonlinear – нелинейные блоки.

Signals & Systems – сигналы и системы.

Sinks - регистрирующие устройства.

Sources — источники сигналов и воздействий.

Subsystems – блоки подсистем.

Список разделов библиотеки Simulink представлен в виде дерева, и правила работы с ним являются общими для списков такого вида:

Пиктограмма свернутого узла дерева содержит символ "+", а пиктограмма развернутого содержит символ "-".

Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме левой клавишей мыши (ЛКМ).

При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (рисунок 4).

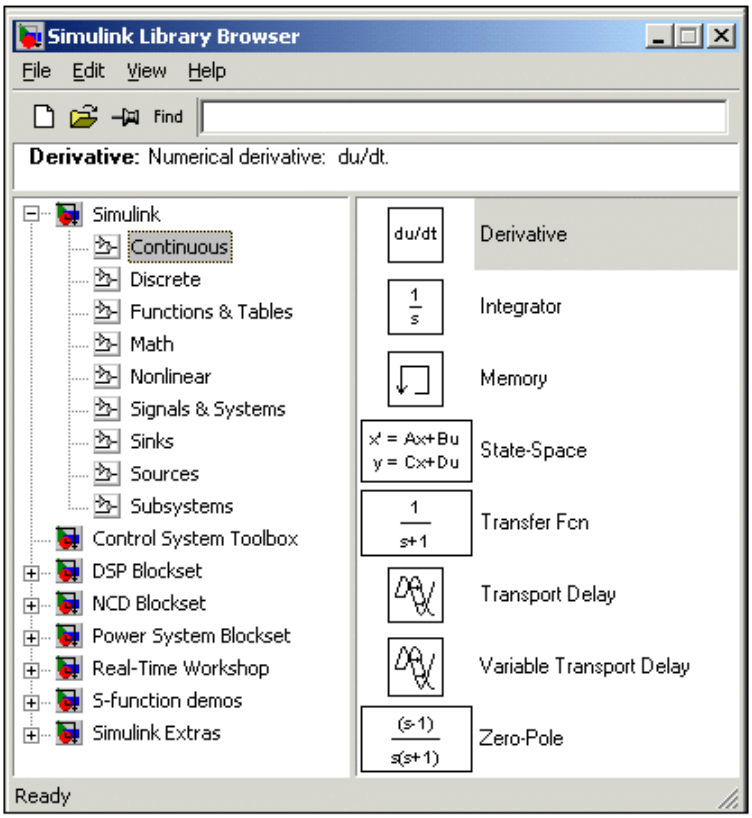


Рисунок 4. Окно обозревателя с набором блоков раздела библиотеки

Для работы с окном используются команды собранные в меню. Меню обозревателя библиотек содержит следующие пункты:

File (Файл) — Работа с файлами библиотек.

Edit (Редактирование) — Добавление блоков и их поиск (по названию).

View (Вид) — Управление показом элементов интерфейса.

Help (Справка) — Вывод окна справки по обозревателю библиотек.

Для работы с обозревателем можно также использовать кнопки на панели инструментов (рисунок 5).

Рисунок 4. Панель инструментов обозревателя разделов библиотек

Кнопки панели инструментов имеют следующее назначение:

1 Создать новую S-модель (открыть новое окно модели).

2 Открыть одну из существующих S-моделей.

3 Изменить свойства окна обозревателя. Данная кнопка позволяет установить режим отображения окна обозревателя "поверх всех окон". Повторное нажатие отменяет такой режим.

4 Поиск блока по названию (по первым символам названия). После того как блок будет найден, в окне обозревателя откроется соответствующий раздел библиотеки, а блок будет выделен. Если же блок с таким названием отсутствует, то в окне комментария будет выведено сообщение Not found <имя блока> (Блок не найден).

Перед выполнением имитационного моделирования необходимо предварительно задать параметры расчета. Задание параметров расчета выполняется в панели управления меню Configuraation Parameters. Вид панели управления приведен на Рисунке 6.

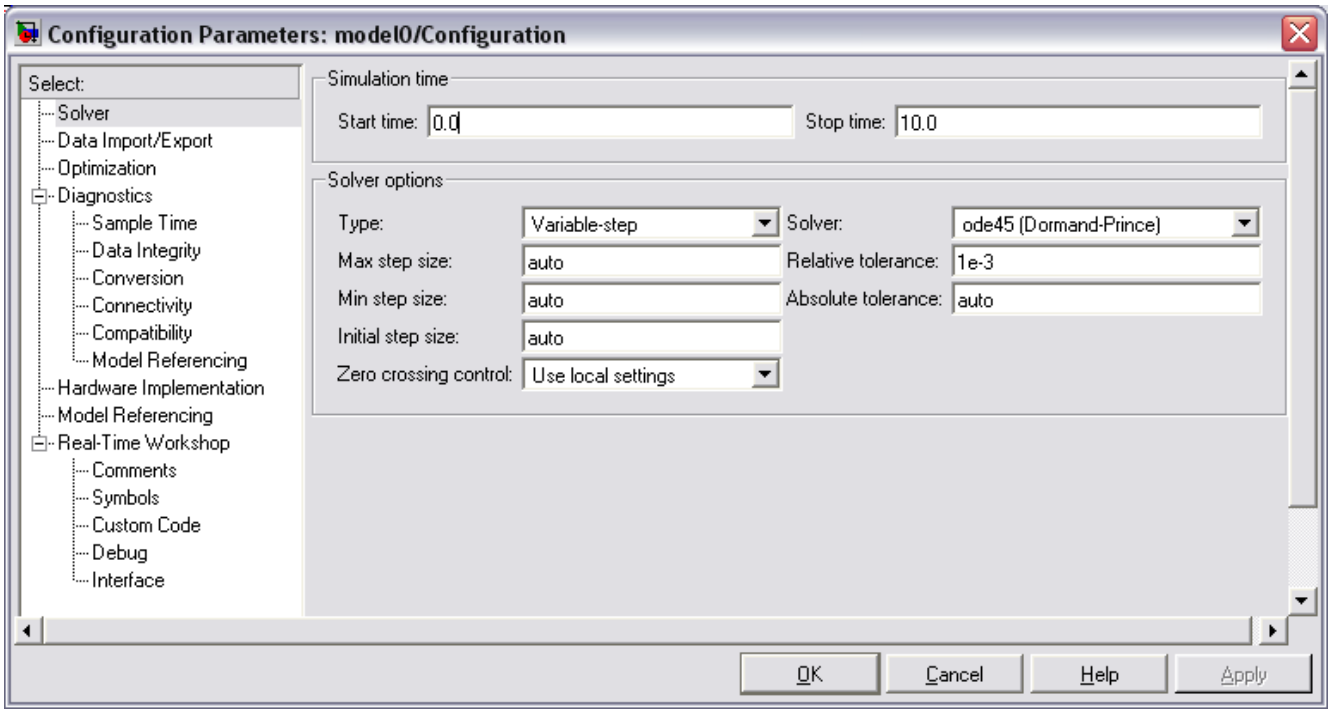


Рис. 6. Панель управления меню Configuraation Parameters.

Окно настройки параметров расчета имеет 4 вкладки:

- Solver (Расчет) — Установка параметров расчета модели.
- Workspace I/O (Ввод/вывод данных в рабочую область) — Установка параметров обмена данными с рабочей областью MATLAB.

- Diagnostics (Диагностика) — Выбор параметров диагностического режима.
- Advanced (Дополнительно) — Установка дополнительных параметров.

Установка параметров расчета модели выполняется с помощью элементов управления, размещенных на вкладке Solver. Эти элементы разделены на три группы (Рисунок 5): Simulation time (Интервал моделирования или, иными словами, время расчета), Solver options (Параметры расчета), Output options (Параметры вывода).

Время расчета (Simulation time) задается указанием начального (Start time) и конечного (Stop time) значений времени расчета. Начальное время, как правило, задается равным нулю. Величина конечного времени задается пользователем исходя из условий решаемой задачи.

При выборе параметров расчета (Solver options) необходимо указать способ моделирования (Type) и метод расчета нового состояния системы. Для параметра Type доступны два варианта - с фиксированным (Fixed-step) или с переменным (Variable-step) шагом. Как правило, Variable-step используется для моделирования непрерывных систем, а Fixed-step - для дискретных.

Список методов расчета нового состояния системы содержит несколько вариантов. Первый вариант (discrete) используется для расчета дискретных систем. Остальные методы используются для расчета непрерывных систем. Эти методы различны для переменного (Variable-step) и для фиксированного (Fixed-step) шага времени, но, по сути, представляют собой процедуры решения систем дифференциальных уравнений.

Ниже двух раскрывающихся списков Type находится область, содержимое которой меняется зависимости от выбранного способа изменения модельного времени. При выборе Fixed-step в данной области появляется текстовое поле Fixed-step size (величина фиксированного шага) позволяющее указывать величину шага моделирования. Величина шага моделирования по умолчанию устанавливается системой автоматически (auto). Требуемая величина шага может быть введена вместо значения auto либо в форме числа, либо в виде вычисляемого выражения (то же самое относится и ко всем параметрам, устанавливаемым системой автоматически).

При выборе Fixed-step необходимо также задать режим расчета (Mode). Для параметра Mode доступны три варианта:

- MultiTasking (Многозадачный) – необходимо использовать, если в модели присутствуют параллельно работающие подсистемы, и результат работы модели зависит от временных параметров этих подсистем. Режим позволяет выявить несоответствие скорости и дискретности сигналов, пересылаемых блоками друг другу.
- SingleTasking (Однозадачный) - используется для тех моделей, в которых недостаточно строгая синхронизация работы отдельных составляющих не влияет на конечный результат моделирования.

- Auto (Автоматический выбор режима) - позволяет Simulink автоматически устанавливать режим MultiTasking для тех моделей, в которых используются блоки с различными скоростями передачи сигналов и режим SingleTasking для моделей, в которых содержатся блоки, оперирующие одинаковыми скоростями.

При выборе Variable-step в области появляются поля для установки трех параметров:

- Max step size - максимальный шаг расчета. По умолчанию он устанавливается автоматически (auto) и его значение в этом случае равно (StopTime — StartTime)/50. Довольно часто это значение оказывается слишком большим, и наблюдаемые графики представляют собой ломаные (а не плавные) линии. В этом случае величину максимального шага расчета необходимо задавать явным образом.
- Min step size - минимальный шаг расчета.
- Initial step size - начальное значение шага моделирования.

В нижней части вкладки Solver задаются настройки параметров вывода выходных сигналов моделируемой системы (Output options). Для данного параметра возможен выбор одного из трех вариантов:

- Refine output (Скорректированный вывод) – позволяет изменять дискретность регистрации модельного времени и тех сигналов, которые сохраняются в рабочей области MATLAB с помощью блока To Workspace. Установка величины дискретности выполняется в строке редактирования Refine factor, расположенной справа. По умолчанию значение Refine factor равно 1, это означает, что регистрация производится с шагом Dt = 1.
- Produce additional output (Дополнительный вывод) — обеспечивает дополнительную регистрацию параметров модели в заданные моменты времени; их значения вводятся в строке редактирования (в этом случае она называется Output times) в виде списка, заключенного в квадратные скобки. При использовании этого варианта базовый шаг регистрации (Dt) равен 1.
- Produce specified output only (Формировать только заданный вывод) — устанавливает вывод параметров модели только в заданные моменты времени, которые указываются в поле Output times (Моменты времени вывода).

Запуск расчета (моделирование) выполняется с помощью выбора пункта меню Simulation/Start. или инструмента  на панели инструментов. Процесс расчета можно завершить досрочно, выбрав пункт меню Simulation/Stop или инструмент  . Расчет также можно остановить (Simulation/Pause) и затем продолжить (Simulation/Continue).

К возможности изменения свойств модели прибегают обычно только опытные пользователи обычно бывает достаточно установок свойств по умолчанию.

Подготовка и запуск модели

- Создание модели
- Моделирование ограничителя
- Основные приемы подготовки и редактирования модели
- Операции форматирования модели

Создание модели
Постановка задачи и начало создания модели

Решение любой проблемы в системе Simulinkдолжно начинаться с постановки задачи. Чем глубже продумана постановка задачи, тем больше вероятность успешного ее решения. В ходе постановки задачи нужно оценить, насколько суть задачи отвечает возможностям пакета Simulink какие компоненты последнего могут использоваться для построения модели.

Основные команды редактирования модели сосредоточены в меню Edit. В качестве примера применения этих команд рассмотрим построение простой модели, а точнее, сразу трех простых моделей в пределах одного окна, можно одновременно моделировать несколько систем.

Сначала откроем пустое окно для новой модели (кнопкаCreateanewmodelв панели инструментов браузера библиотекSimulink).

Ввод текстовой надписи

Введем заголовок нашей будущей модели - «Simplemodel» (Простая модель). Для этого достаточно установить курсор мыши в нужное место окна и дважды щелкнуть левой кнопкой мыши. Появитсяпрямоугольная рамка, внутри которой находится мигающий маркер ввода в виде вертикальной палочки.

Теперь можно ввести нужную надпись по правилам, действующим для строчного редактора. Пока будем считать, что параметры надписи по умолчанию нас вполне устраивают.

Размещение блоков в окне модели

Из раздела библиотеки Sourcesперетащим мышью три источника сигнала: синусоидального, прямоугольного (дискретного) и пилообразного. Рис 7,8,9

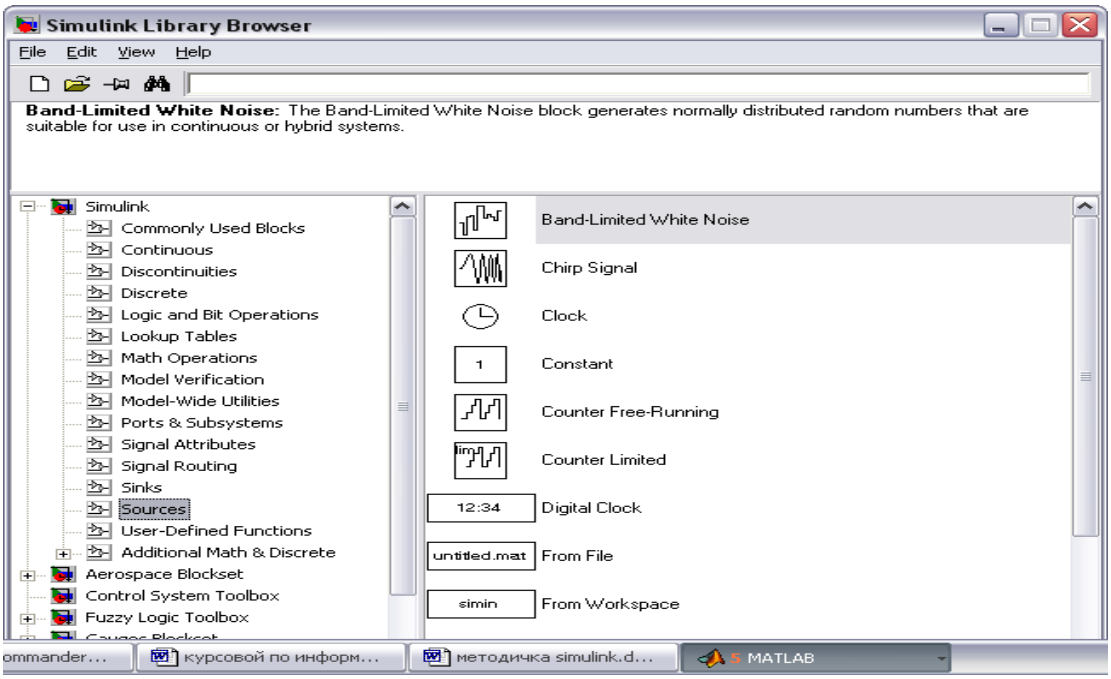


Рис.7. Источники сигналов

Затем из раздела Sinksперетащим в окно модели блок осциллографа.

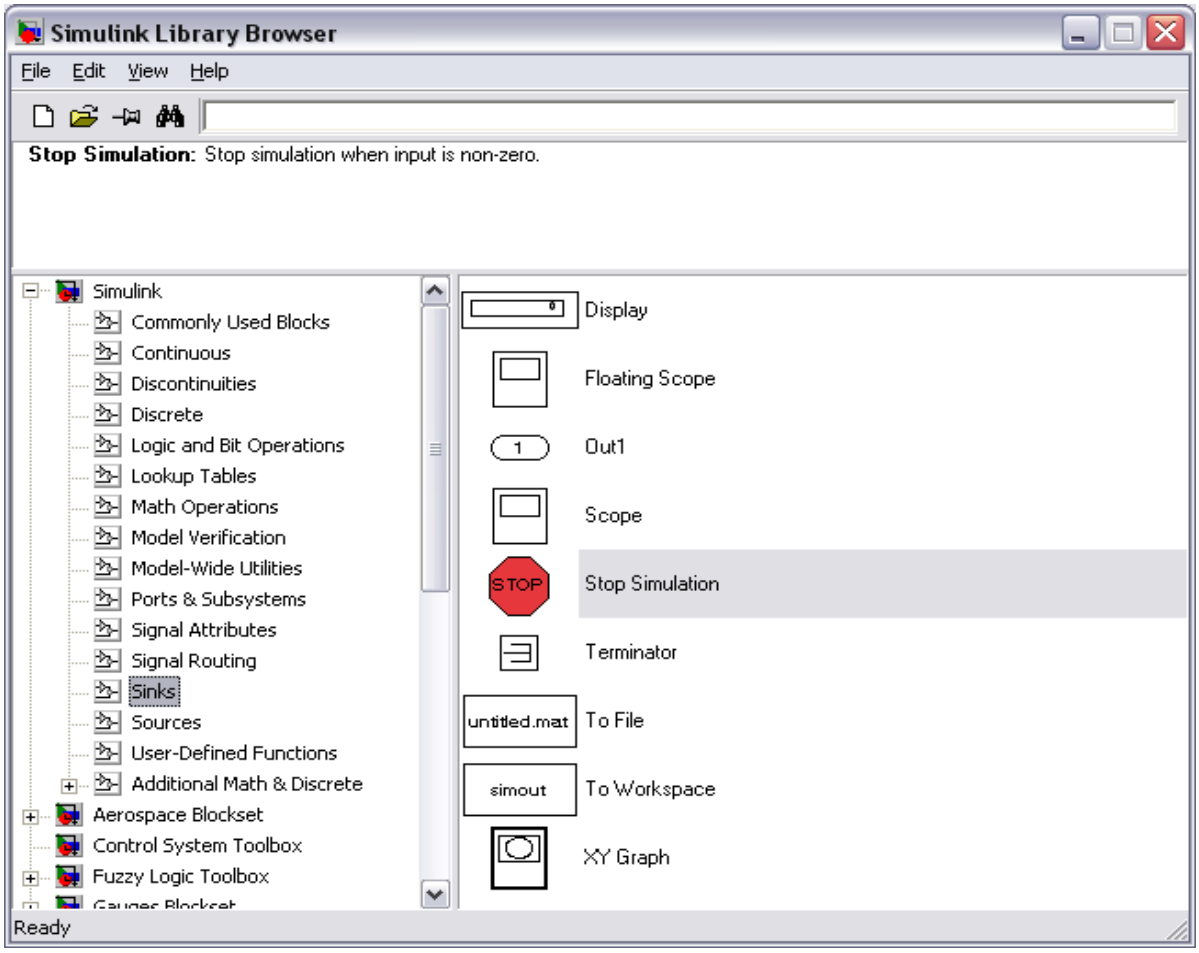


Рис.8. Источники сигналов

В результате получим модель в виде, представленном на рис.8.

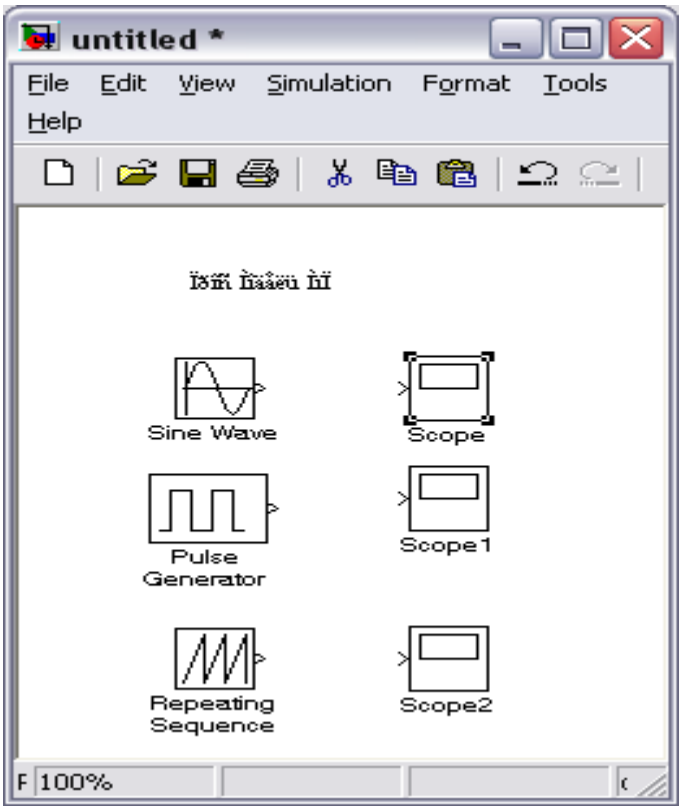


Рисунок . 9. Исотчники сигналов

Выделение блока модели

На рисунке 6 показано также меню редактированияEditв открытом виде – при выделении блока в этом меню становятся доступны команды редактирования свойств блока. Для выделения блока достаточно навести на него маркер мыши и нажать левую кнопку. В рамке блока по углам появятся маленькие темные прямоугольники, которые и являются признаком того, что блок выделен. На рисунке 6 выделен блок осциллографаScope.

Если захватить курсором мыши уголок выделенного блока, то можно заметить, что курсор мыши превратится в перекрестие тонких диагональных двухсторонних стрелок. Это означает, что можно пропорционально увеличивать или уменьшать блок в диагональных направлениях.

Меню редактирования Edit

Кратко рассмотрим основные команды меню Edit(рисунок 6). Это меню содержит ряд типовых команд, которые разбиты на 6 групп. В первой группе есть две команды:Undo(отмена последней операции) и Redo(восстановление последней отмененной операции). Эти команды являются контекстно-зависимыми.

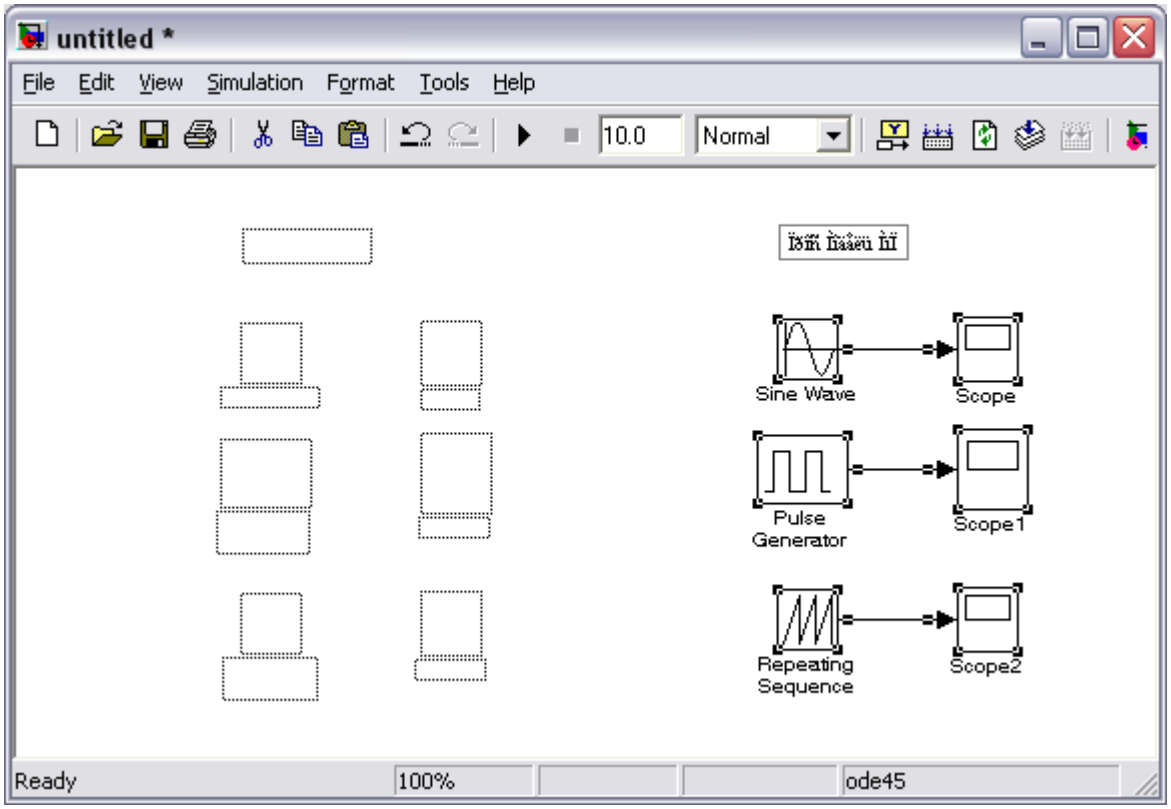
Следующая группа команд связана с операциями с буфером обмена Windows:

- Cut- перенос выделенных объектов в буфер;
- Copy - - копирование выделенных объектов в буфер;
- Paste- вставка объектов из буфера в заданное курсором мыши место;

- Clear- уничтожение выделенных объектов;
- SelectAll- - выделение всех объектов модели;
- Copymodeltoclipboardкопирование всей модели в буфер;
- Find- - поиск в модели заданного объекта.

Остальные команды подменю Editносят специальный характер.

Теперь можно приступить к соединению выходов источников со входами осциллографов. Для этого достаточно указать курсором мыши на начало соединения (выход источника) и затем при нажатой левой кнопке мыши протянуть соединение в его конец (вход осциллографа). В итоге получим модель, показанную на рисунке. 10.



Рисунке 10. Готовая модель

4 Порядок выполнения практической работы

1 Запустить Simulink.

2 Изучить основную библиотеку Simulink. Открыть и изучить все разделы библиотеки.

3 В библиотеке Simulink вызвать DEMOS.

4 Изучить простые модели

6 Контрольные вопросы

1 Расскажите об основных разделах библиотеки.

2 Что входит в раздел Continuous?

3 Что входит в раздел Discrete?

4 Что входит в раздел Functions & Tables?

5 Что входит в раздел Math?

6 Что входит в раздел Signals & Systems?

7 Что входит в раздел Sinks?

8 Что входит в раздел Sources?

9 Какие результаты получены при моделировании прыгающего шара?

3. Практическая работа 3. Математические модели на основе обыкновенных дифференциальных уравнений (ОДУ) (8 часов)

Цель работы: научиться составлять схемы решения систем обыкновенных дифференциальных уравнений (ОДУ) в среде Simulink пакета MatLab.

1. Решение ОДУ первого порядка.

Основой для решения обыкновенных дифференциальных уравнений первого порядка является задача Коши:

$$\frac{dy}{dx} = f(x, y)$$

с одной зависимой переменной y(x).

Пример.

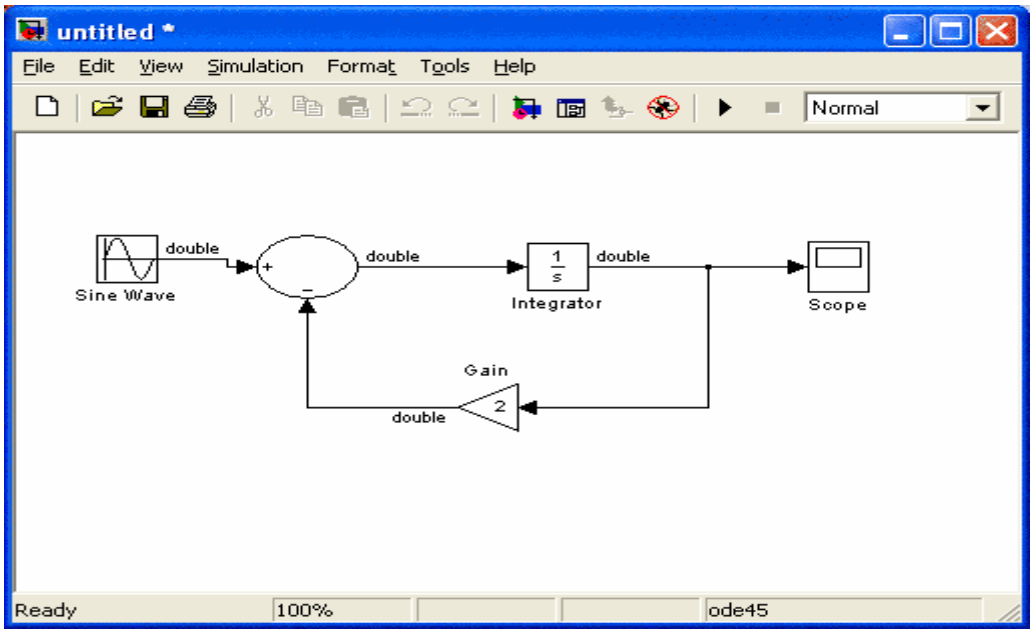
Дано дифференциальное уравнение

x'(t) + 2x(t) = sin(t),

x(0) = 0.

После запуска системы MatLab нажмем кнопку Simulink, а затем в открывшемся окне кнопку Create a new Model. В открывшемся файле создадим схему решения уравнения, перетаскивая при нажатой левой кнопки мыши необходимые блоки из окна Simulink Library Brouser.

Для построения схемы решения уравнения в Simulink используется блок Integrator (класс Continuos). На его вход подается производная, а на выходе получают величину x. Блоки Sum (Сумматор) и Gain (Усилитель) (класс Math) необходимы для формирования значения x' в соотствии с ОДУ. Для получения сигнала sin(t) используется блок Sine Wave (класс Sources), в котором необходимо провести установки, соответствующие задаче, открыв блок двойным щелчком мыши или выбрав опцию Block Parameters при нажатой правой кнопке мыши. Полученное значение x(t) подается на вход блока Scope. При открытии данного блока появляется график решения. Установить масштабы осей, соответствующие полученному решению можно, нажав кнопку Autoscale. Рис.11



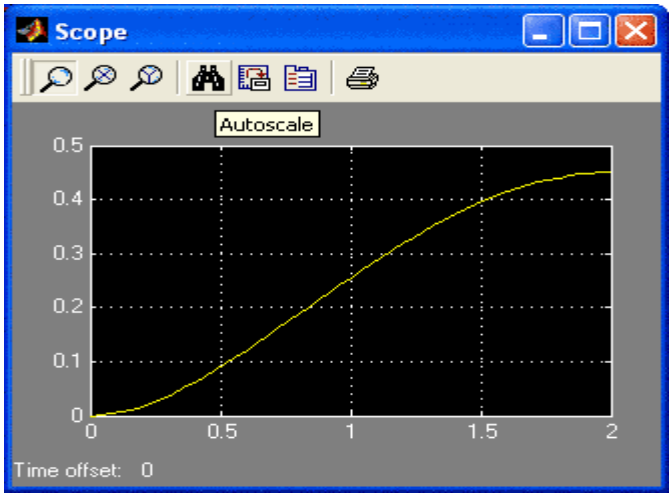
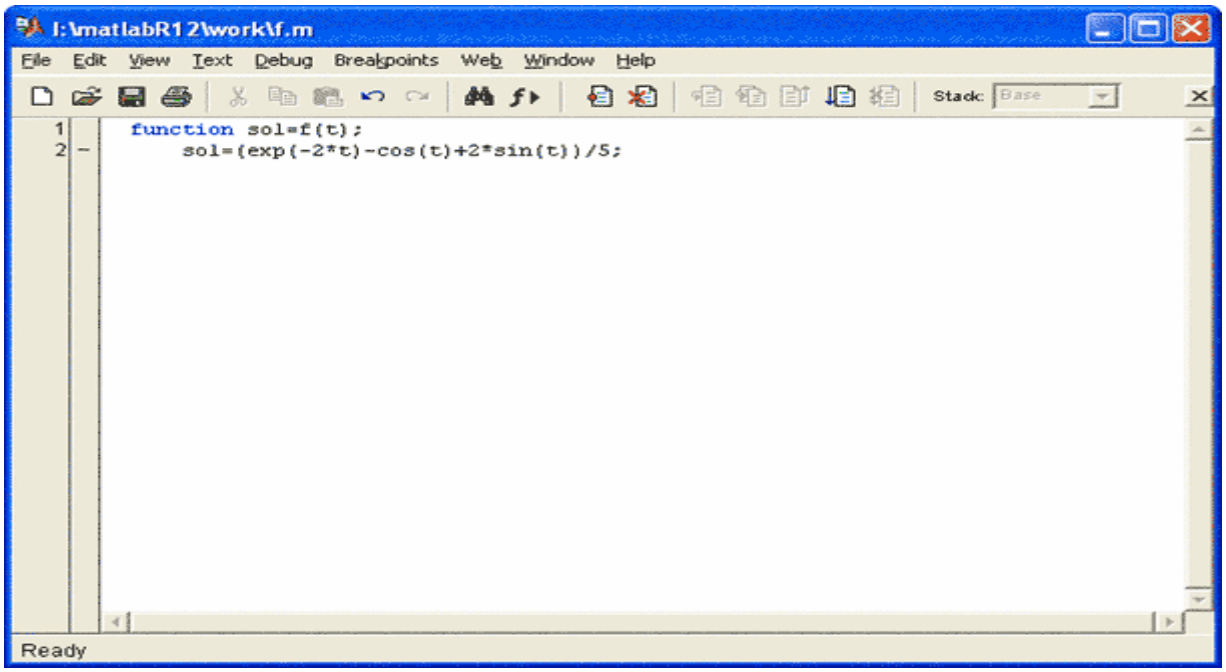


Рис.11. Модель и график решения

Для проверки найденного решения в окне Command Window создадим M - файл для решения задачи (File ==> New ==> M-file). В открывшемся окне создадим функцию решения задачи, которую сохраним в текущей директории под именем f.m (указанное имя система предлагает по умолчанию).



После этого в командном окне наберем текст:

```
>> t=(0:0.1:2);
>> y=f(t);
>> plot(t,y)
>> grid on
```

После выполнения команд открывается окно с графиком функции. Очевидно, что два полученных графика идентичны.Рис.12.

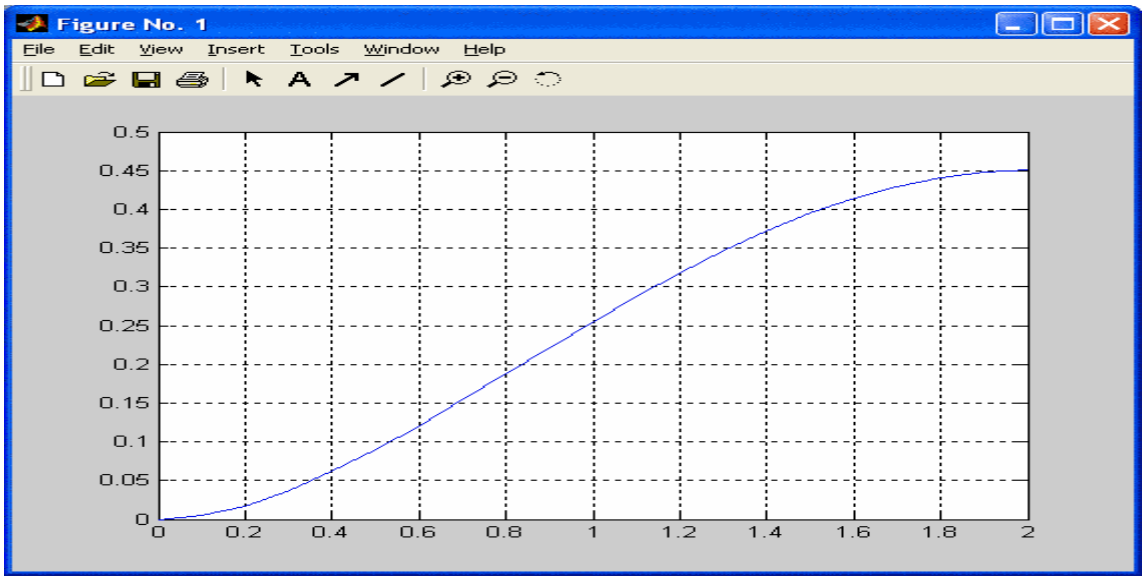


Рис.12. Окно с графиком

2. Решение систем ОДУ первого порядка.

Рассмотрим решение системы ОДУ первого порядка.

Пример.

Модифицированная задача Лотки – Вольтера (модель хищник-жертва).

С учетом самоограничения на рост популяции жертв

$$\begin{cases} \frac{dx}{dt} = x(\alpha - \beta y - \gamma x), \\ \frac{dy}{dt} = -y(\delta - \varepsilon x). \end{cases}$$

Зададим параметры задачи: α = 0.1; β = 0.05; γ = 0.03; δ = 0.2; ε = 0.15.

Структурная схема решения задачи в системе Simulink:Рис.13

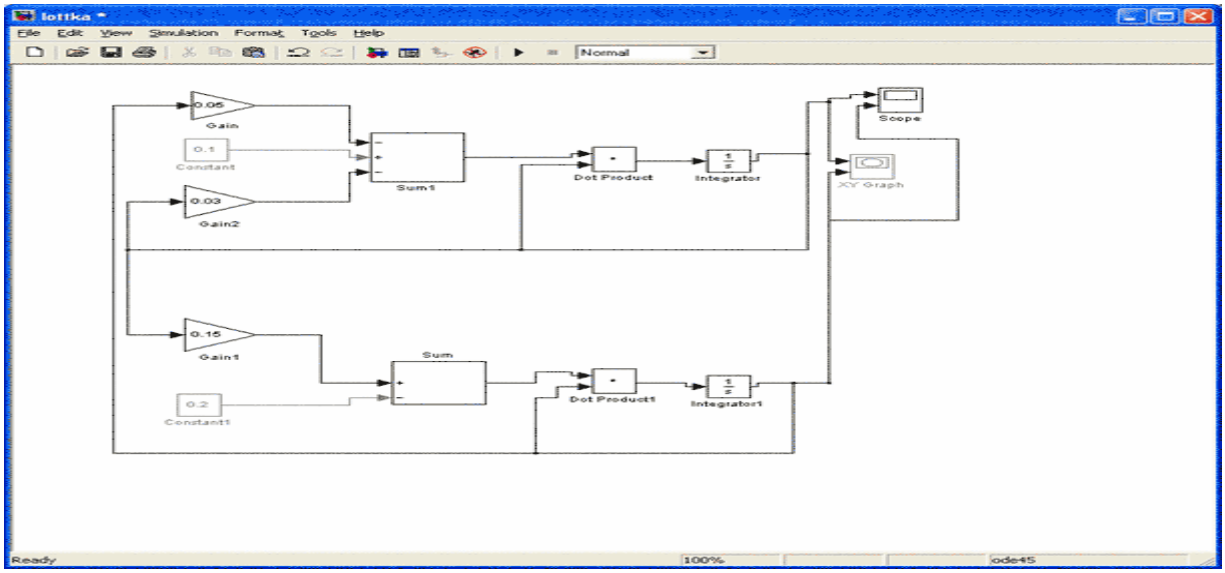


Рис.13. Визуальная модел оешения

Раскрыв блок интегратора, зададим начальные значения: $y_1 = 2$, $y_2 = 0.01$.

После окончания моделирования, раскрывая блоки Scope и XY-Graph, можно увидеть графики изменения численности решения системы, Рис.14.

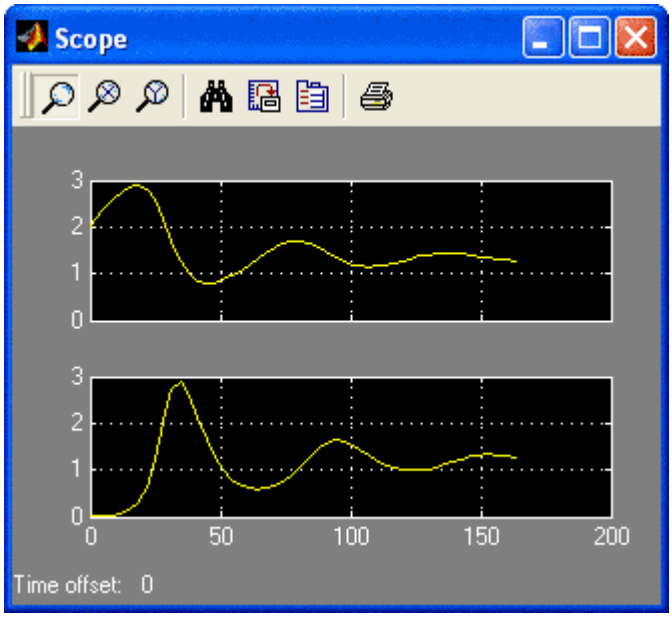


Рис.14. График изменения численности

Создадим также в MatLab M - файл для задания правой части системы ОДУ:

```
function dy=vlm(t,y)
dy=zeros(2,1);
dy(1)=y(1)*(0.1-0.05*y(2)-0.03*y(1));
dy(2)=-y(2)*(0.2-0.15*y(1));
```

Для конкретного набора начальных значений систему можно решить, используя метод Рунге - Кутты 4-го порядка (встроенная функция ode45):

```
> [T, Y] = ode45('vlm',[0 164],[2 0.01]);
```

Параметры функции ode45: имя M - файла, диапазон изменения независимой переменной, начальные значения.

Задание.

1. Построить схемы решения рассмотренных задач в системе Simulink, получить графики решения. Сравнить с решением задач в MatLab с помощью функции ode45.

2. Решить эти же задачи в MatLab, построить графики решения.
3. Построить схему решения в Simulink и получить графики решения следующих задач.

Контрольные вопросы

1. В чем суть дифференциального уравнения
2. Что такое ОДУ
3. В чем недостаток уравнения Эйлера
4. Зачем вводится постоянная составляющая CONST
5. Зачем нужны начальныя условия

4. Практическое занятие 4. Методология структурного синтеза моделей (8 часов)

Системное проектирование моделей – это методика, формирующая компоненты и способы их соединения, задающая ограничения, при которых система должна функционировать, выбирающая наиболее эффективное сочетание людей, элементов и программного обеспечения для реализации системы. SADT (Structured Analysis & Design Technique – методология структурного анализа и проектирования) – одна из самых известных и широко используемых систем проектирования, разработанная более 30 лет назад.

Некоторые из областей эффективного применение SADT: программное обеспечение сетей, системная поддержка и диагностика, долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, конфигурация компьютерных систем, обучение персонала, встроенное программное обеспечение для оргсисте.

Полезность технологии SADT привела к стандартизации ее части, называемой IDEF0.

Универсальной единицей в данной технологии является SA-блок, Рис.15:

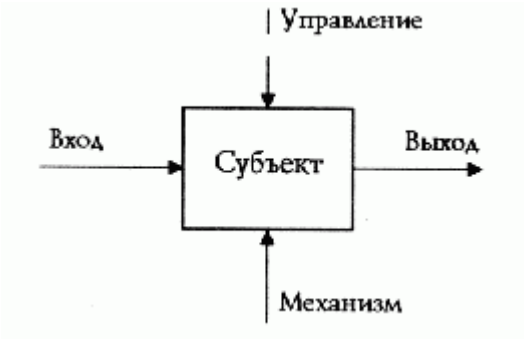


Рис. 15. SA-блок

Вход при наличии управления преобразуется в выход с помощью «механизма» (исполнители и ресурсы).

Выходы одного блока могут быть входами или управлениями (или исполнителями) для других блоков. (Блоки именуются, а дуги помечаются с использованием естественного языка.) Дуги могут разветвляться и соединяться, а каждый блок может быть подвергнут декомпозиции, т. е. разделен как целое на свои составляющие на более детальной диаграмме. Входы, управления и выходы определяют интерфейсы между блоками, а исполнители позволяют при необходимости в определенной степени объединять объекты. Границы блоков и диаграмм должны быть согласованы, а возникающая иерархическая, взаимосвязанная совокупность диаграмм является моделью.

Диаграмма ограничивается 3-6 блоками для того, чтобы детализация осуществлялась постепенно. Вместо одной громоздкой модели используется несколько небольших взаимосвязанных моделей, значения которых взаимно дополняют друг друга, делая понятной структуризацию сложного объекта.

Во одних случаях важно рассмотрение динамики системы, а в других требуется получить распределенную базу знаний. Ориентация модели (ее контекст, точка зрения и цель) может быть направлена так, что результирующие структурные описания дадут исходные данные для методологий имитационного моделирования, проектирования баз данных

или структурного программного проектирования. Существует два основных направления в SA-моделировании: функциональные модели выделяют события в системе, модели данных выделяют объекты системы, которые связывают функции между собой и с их окружением. В обоих случаях используется один и тот же графический язык блоков и дуг (хотя это использование двойственно: блоки и дуги меняются ролями).
С точки зрения SADT модель может быть сосредоточена либо на функциях системы, либо на ее объектах. SADT-модели, ориентированные на функции, принято называть функциональными моделями, а ориентированные на объекты системы – моделями данных, Функциональная модель представляет с требуемой степенью детализации систему функций, которые в свою очередь отражают свои взаимоотношения через объекты системы.

Модели данных дуальны к функциональным моделям и представляют собой подробное описание объектов системы, связанных системными функциями. Полная методология SADT поддерживает создание множества моделей для более точного описания сложной системы.

Целью модели является получение ответов на некоторую совокупность вопросов. Эти вопросы неявно присутствуют (подразумеваются) в процессе анализа и, следовательно, они руководят созданием модели и направляют его. Это означает, что сама модель должна будет дать ответы на эти вопросы с заданной степенью точности. Если модель отвечает не на все вопросы или ее ответы недостаточно точны, то мы говорим, что модель не достигла своей цели. Определяя модель таким образом, SADT закладывает основы практического моделирования.

Обычно вопросы для SADT-модели формулируются на самом раннем этапе проектирования, при этом основная суть этих вопросов должна быть выражена в одной-двух фразах. На рис. 1 показана модел,

использующая SADT для определения цели модели экспериментального механического цеха (ЭМЦ).

Поскольку модель будет использована для подготовки учебного руководства, разумная степень точности будет достигнута, если каждая описанная в модели функция экспериментального цеха будет изложена в одном абзаце текста. Такая точность достижима и измерима. Другие методы анализа систем (альтернативные пути описания системы) не учитывают этот критический момент определения основной цели модели. Только поняв, насколько хорошо нужно ответить на поставленные вопросы, можно определить,

когда процесс моделирования можно считать завершенным (т.е. когда модель будет соответствовать поставленной цели).

С определением модели тесно связана позиция, с которой наблюдается система и создается ее модель. Поскольку качество описания системы снижается, если оно не сфокусировано ни на чем, SADT требует, чтобы модель рассматривалась все время с одной и той же позиции. Эта позиция называется «точкой зрения» данной модели.

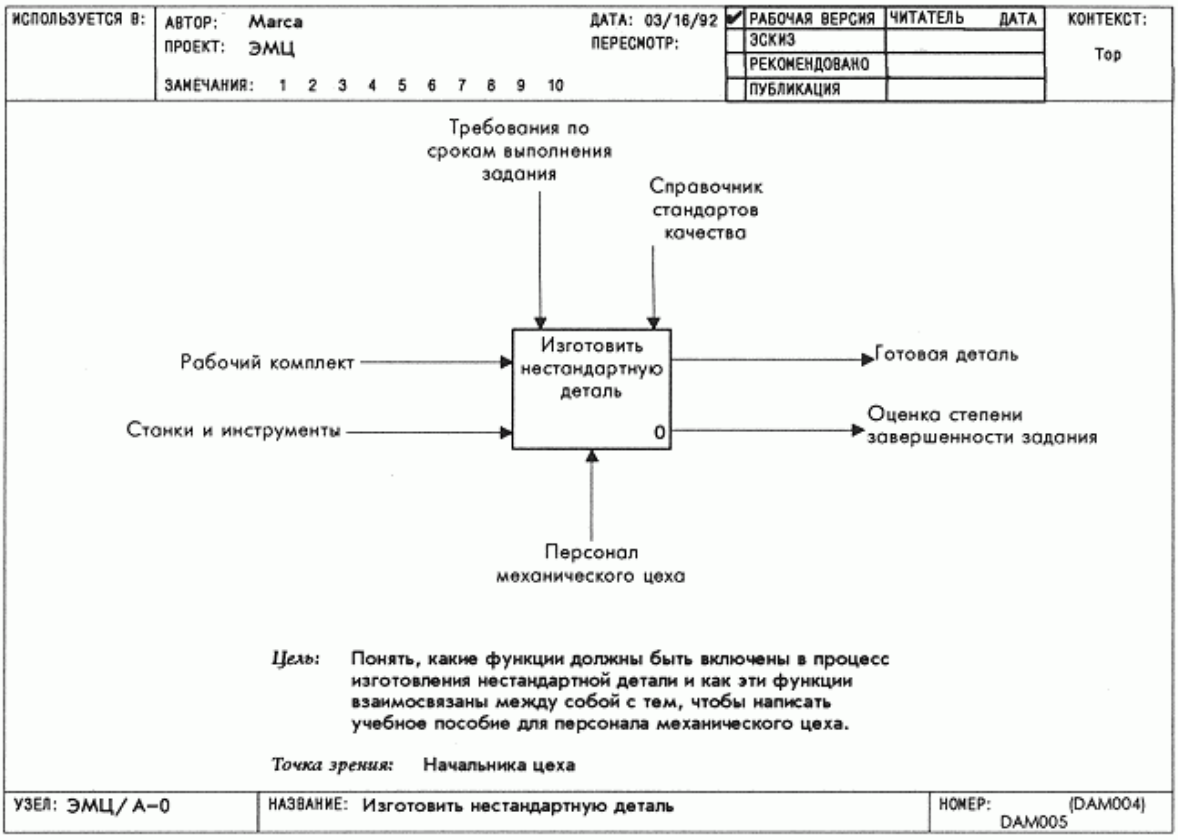
SADT-модель объединяет и организует диаграммы в иерархические структуры, в которых диаграммы наверху модели менее детализированы, чем диаграммы нижних уровней. Другими словами, модель SADT можно представить в виде древовидной структуры диаграмм, где верхняя диаграмма является наиболее общей, а самые нижние наиболее детализированы. На рис. 1 представлены две диаграммы из модели экспериментального механического цеха. Верхняя диаграмма (на вершине модели) описывает механический цех как функцию, в основе которой лежит преобразование входящих рабочих комплектов (заготовок, сырья, документации) в детали при определенном контроле качества. Нижняя диаграмма детализирует верхнюю, указывая на три главные функции механического цеха: управление выполнением заданий,

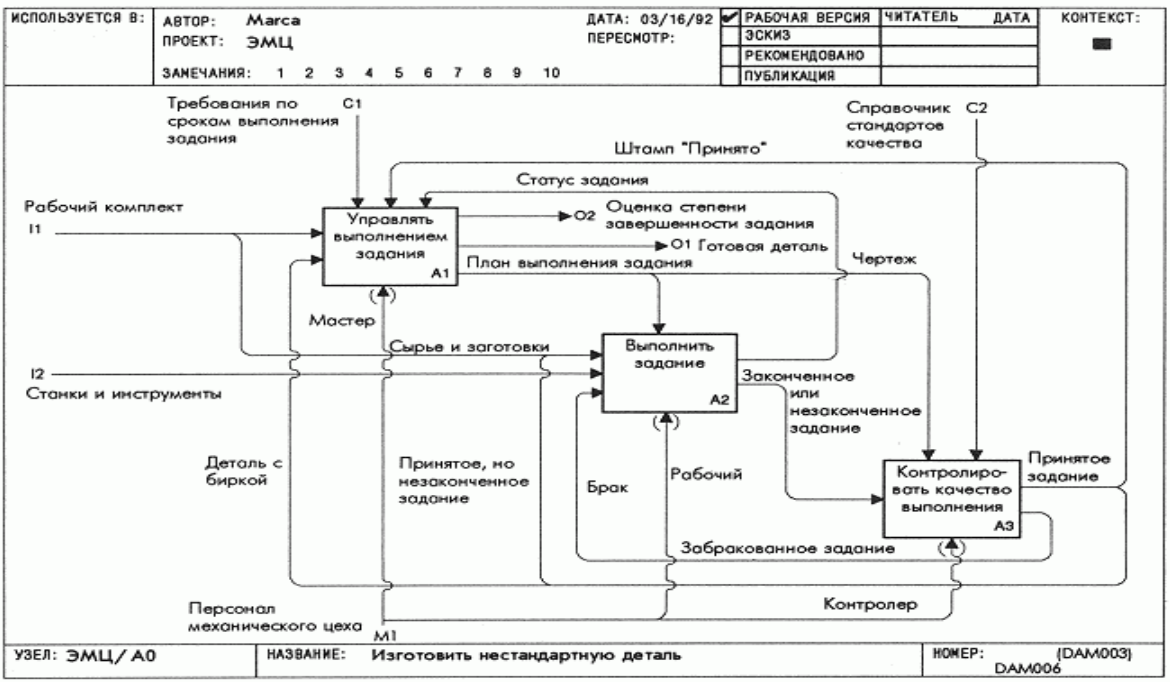
выполнение задания и контроль качества выполнения. Таким образом, общая функция, указанная на верхней диаграмме, детализируется с помощью трех функций на нижней диаграмме.

Взаимное влияние трех функций нижней диаграммы, обозначенное дугами, которые символизируют объекты механического цеха. Имена этих дуг совпадают с теми, что указаны на дугах верхней диаграммы. Это пример того, как SADT соединяет диаграммы в модели через объекты системы. Такая схема соединения требует согласованного наименования и учета объектов системы с тем, чтобы две диаграммы можно было рассматривать как связанные между собой. Например, функциональный блок на верхней диаграмме имеет семь дуг, и каждая из них может быть найдена среди дуг, идущих к границе или от границы диаграммы на следующем уровне.

Каждая SADT-диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними. Диаграмме дается название, которое располагается в центре нижней части ее бланка. На каждой диаграмме написана стандартно идентифицирующая ее информация: автор диаграммы, частью какого проекта является работа, дата создания или последнего пересмотра диаграммы, статус диаграммы. Вся идентифицирующая информация располагается в верхней части бланка диаграммы.

Функциональные блоки на диаграммах изображаются прямоугольниками. Блок представляет функцию или активную часть системы, поэтому названиями блоков служат глаголы или глагольные обороты. Например, названиями блоков диаграммы выполнить задание являются: определить степень выполнения задания, выбрать инструменты, подготовить рабочее место, обработать на станке и собрать, как показано на рис. 16.


ИСПОЛЬЗУЕТСЯ В:
АВТОР: Marca
ПРОЕКТ: ЭМЦ
ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10
ДАТА: 03/16/92
ПЕРЕСМОТР:
РАБОЧАЯ ВЕРСИЯ
ЭСКИЗ
РЕКОМЕНДОВАНО
ПУБЛИКАЦИЯ
ЧИТАТЕЛЬ ДАТА
КОНТЕКСТ:
Top
Требования по срокам выполнения задания
Справочник стандартов качества
Рабочий комплект
Станки и инструменты
Изготовить нестандартную деталь
0
Готовая деталь
Оценка степени завершенности задания
Персонал механического цеха
Цель: Понять, какие функции должны быть включены в процесс изготовления нестандартной детали и как эти функции взаимосвязаны между собой с тем, чтобы написать учебное пособие для персонала механического цеха.
Точка зрения: Начальника цеха
УЗЕЛ: ЭМЦ/ A-0
НАЗВАНИЕ: Изготовить нестандартную деталь
НОМЕР: (DAM004) DAM005


ИСПОЛЬЗУЕТСЯ В:
АВТОР: Marca
ПРОЕКТ: ЭМЦ
ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10
ДАТА: 03/16/92
ПЕРЕСМОТР:
РАБОЧАЯ ВЕРСИЯ
ЭСКИЗ
РЕКОМЕНДОВАНО
ПУБЛИКАЦИЯ
ЧИТАТЕЛЬ ДАТА
КОНТЕКСТ:
Требования по срокам выполнения задания
C1
Справочник стандартов качества
C2
Штамп "Принято"
Статус задания
Рабочий комплект
I1
Управлять выполнением задания
A1
O2 Оценка степени завершенности задания
O1 Готовая деталь
План выполнения задания
Чертеж
Мастер
Сырье и заготовки
I2
Станки и инструменты
Выполнить задание
A2
Законченное или незаконченное задание
Деталь с биркой
Принятое, но незаконченное задание
Брак
Рабочий
Контролировать качество выполнения
A3
Принятое задание
Забракованное задание
Персонал механического цеха
M1
Контролер
УЗЕЛ: ЭМЦ/ A0
НАЗВАНИЕ: Изготовить нестандартную деталь
НОМЕР: (DAM003) DAM006

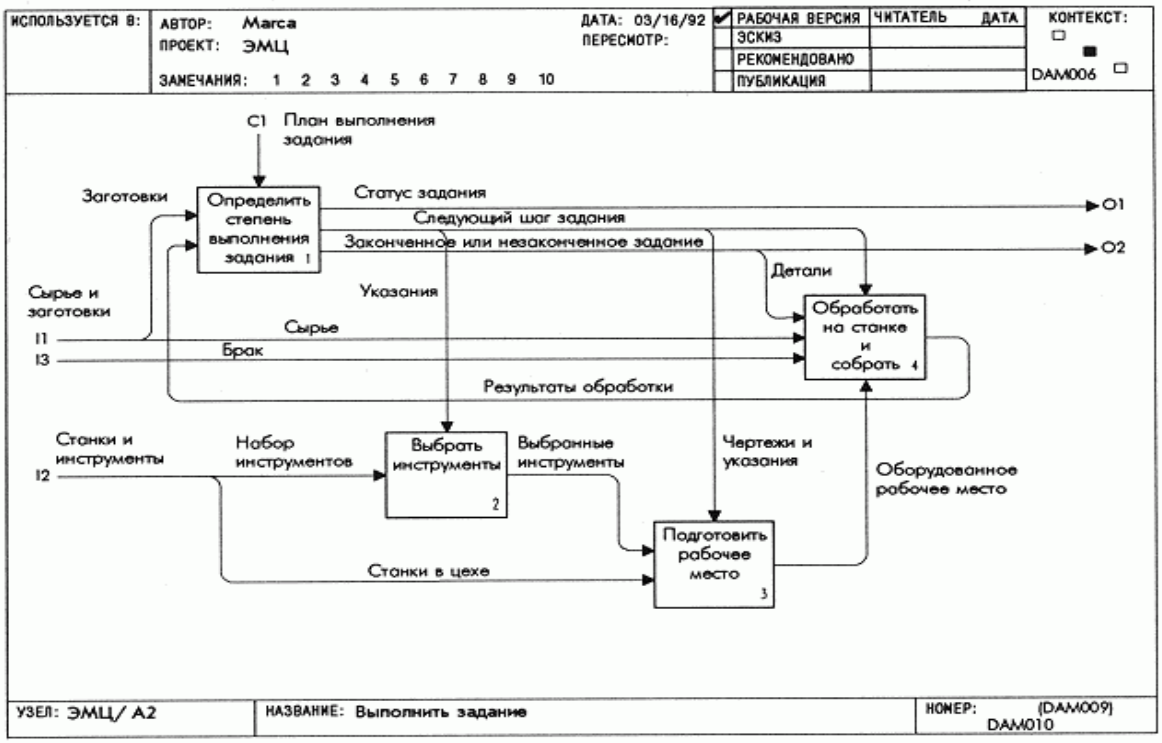


Рис.16. Диаграммы содержат блоки и дуги

Кроме того, SADT требует, чтобы в диаграмме было не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, доступном для чтения, понимания и использования. Другими словами, SADT-диаграммы и SADT-модели наглядны.

В отличие от других графических методов структурного анализа в SADT каждая сторона блока имеет особое, вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя – для управления, правая – для выходов, нижняя – для механизмов. Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, кто, что и как выполняет функция. Например, четвертый блок диаграммы выполнить задание может быть интерпретирован следующим образом:

детали, сырье и брак, обрабатываются на станке и собираются в результаты обработки с использованием оборудованного рабочего места. Блоки SADT не размещаются на диаграмме случайным образом, а они размещаются по степени важности, как ее понимает автор диаграммы. В SADT этот относительный порядок называется доминированием. Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы. Например, самым доминирующим блоком диаграммы может быть либо первый из требуемой последовательности функций, либо планирующая или контролирующая функция, влияющая на все.

Дуги на SADT-диаграмме изображаются одинарными линиями со стрелками на концах. Для функциональных SADT-диаграмм дуга представляет множество объектов. Нужно использовать общее понятие «объекты», поскольку дуги в SADT могут представлять, например, планы, данные в компьютерах, машины и информацию. Дуги диаграммы выполнить задание на рис. 2 представляют материалы, написанные на бумаге (например, следующий шаг задания), физические материалы (например, сырье и заготовки), инструменты (например, набор инструментов), рабочие чертежи (например, чертежи и указания), рабочую среду (например, оборудованное рабочее место) и управленческую информацию (например, статус задания). Однако в системном анализе вместо термина «объекты» часто употребляют термин «данные». Это объясняется тем, что системному анализу ранее подвергались, как правило, системы программного обеспечения.

Так как в SADT дуги изображают объекты, они описываются (помечаются) существительными или существительными с определениями, располагающимися достаточно близко к линии дуги. Рекомендуется размещать описания дуг, называемые метками, как можно ближе к линиям дуг, не нарушая, однако, читабельность диаграмм. Это

устраняет неопределенность в том, к какой дуге относится метка, и исключается необходимость в дополнительных графических связях (например, в «зигзагах»). Все метки дуг на диаграмме выполнить задание расположены вплотную к соответствующим дугам. Рекомендуется принять этот стиль описания дуг для того, чтобы диаграммы были упорядоченными и простыми для чтения.

Между объектами и функциями возможны четыре отношения: вход, управление, выход, механизм. Каждое из этих отношений изображается дугой, связанной с определенной стороной блока. По соглашению левая сторона блока предназначена для входных дуг, верхняя сторона – для управленческих дуг, правая сторона – для выходных дуг, нижняя сторона – для дуг механизмов. Таким образом, стороны блока чисто графически сортируют объекты, изображаемые касающимися блока дугами.

Входные дуги изображают объекты, используемые и преобразуемые функциями. Например, в процессе изготовления детали сырье трансформируется функцией обработать на станке и собрать. Управленческие дуги представляют информацию, управляющую действиями функций. Обычно управляющие дуги несут информацию, которая указывает, что должна выполнять функция. Например, следующий шаг задания определяет, какие нужно выбрать инструменты, какие потребуются станки и цеха и как инструменты и станки должны использоваться при изготовлении детали. Выходные дуги изображают объекты, в которые преобразуются входы. Например, обработать на станке и собрать преобразует сырье и брак в результаты обработки, которые в конечном итоге становятся деталями. Дуги механизмов отражают реализацию функции системы). Например, подготовить рабочее место организует инструменты и станки в эффективное пространство для следующего шага задания. Это – рабочая среда, называемая оборудованным рабочим местом. Она обозначает место, где рабочий изготавливает деталь, реализуя функцию обработать на станке и

собрать. Таким образом, механизмы изображают физические аспекты функции (склады, людей, организации, приборы).

SADT-диаграмма составлена из блоков, связанных дугами, определяющих, как блоки влияют друг на друга. Это влияние может выражаться либо в передаче выходной информации к другой функции для дальнейшего преобразования, либо в выработке управляющей информации, предписывающей, что именно должна выполнять другая функция. Например, блок обработать на станке и собрать влияет на блок определить степень выполнения задания, выдавая ему результаты обработки для оценки, а блок определить степень выполнения задания влияет на очередную операцию блока обработать на станке и собрать с помощью следующего шага задания. Другими словами, существует сильная управляющая связь блока определить степень выполнения задания с блоком обработать на станке и собрать и наряду с ней более слабая связь по входу-выходу от блока обработать на станке и собрать к блоку определить степень выполнения задания. SADT-диаграммы предназначены для представления входных-выходных преобразований и указывающие правила этих преобразований. Дуги на них изображают интерфейсы между функциями системы, а также между системой и ее окружающей средой.

Достоинства и недостатки SADT-моделей

Преимущество построенных моделей в том, что их легко обсуждать с экспертами предметной области в силу их наглядного графического представления, а кроме того, основные системные понятия формируют словарь предметной области, который закладывается в информационную систему. IDEF-диаграмма представляет собой статическую модель, и проверить ее поведение, динамические характеристики невозможно. В ней не представлены в явном виде объекты системы и их атрибуты, значения которых изменяются в результате осуществления процессов и

влияют на дальнейшее функционирование системы, а также формируют потоки данных в системе.

SADT-диаграмма содержит от трех до шести блоков, связанных дугами, и имеет при построении модели несколько версий. Для того чтобы различать версии одной и той же диаграммы, используются С-номера. Блоки на диаграмме изображают системные функции, а дуги изображают множество различных объектов системы. Блоки обычно располагаются на диаграмме в соответствии с порядком их важности относительно друг друга. Дуги, связывающие блоки, изображают наборы объектов и могут разветвляться и соединяться различными сложными способами. Однако, разветвляясь и соединяясь, дуги должны во всех случаях сохранять представляемые ими объекты.

Процесс моделирования может быть разделен на несколько этапов: опрос экспертов, создание диаграмм и моделей, распространение документации, оценка адекватности моделей и принятие их для дальнейшего использования.

Задание

Необходимо составить 1 диаграмму 0-го уровня, 1 диаграмму 1-го уровня и минимум 2 диаграммы второго уровня, содержащие модели в формате IDEF0. Если тема своей работы отсутствует, то предлагаются на выбор следующие варианты заданий

1. Вычисление интеграла.
2. Написание программы.
3. Сдача долгов по экзаменационной сессии.
4. Поступление на работу.
5. Установка операционной системы на компьютер.
6. Обучение в ВУЗе
7. Бизнес планирование

Контрольные вопросы

1. Что такое бизнес-процесс?
2. Какие типы моделирования информационных систем Вы знаете?
3. Что такое IDEF0?
4. Для чего был разработан стандарт IDEF0?
5. Какие правила используются для нотаций IDEF0?
6. Какие основные определения используются в нотации IDEF0?
7. Какие различают два класса моделей по отношению к реально существующему бизнес-процессу?

5. **Практическая работа 5. Пакет ANYLOGIC для моделирования сложных систем (8 часов)**

Цель практической работы Изучить методологию агентного моделирования. Приобрести практические навыки работы с системой AnyLogic при построении агентных моделей.

Порядок выполнения работы Агенты в AnyLogic Агент – это некоторая сущность, обладающая активностью, автономным поведением, принимающая решения в соответствии с некоторым набором правил, взаимодействующая с окружением и другими агентами, а также сама изменяющаяся. 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. При помощи агентов моделируют рынки (агент – потенциальный покупатель), конкуренцию и цепочки поставок (агент – компания), население (агент – семья, житель города или избиратель) и мн. др.

В среде AnyLogic можно легко и быстро создавать модели с агентами. Агент естественно реализовывать с помощью базового элемента AnyLogic – активного объекта. В модели можно создавать классы активных объектов и далее использовать в модели любое число экземпляров этих классов. Активный объект имеет параметры, которые можно изменять извне, переменные, которые можно считать памятью агента, а также поведение (рис. 17).

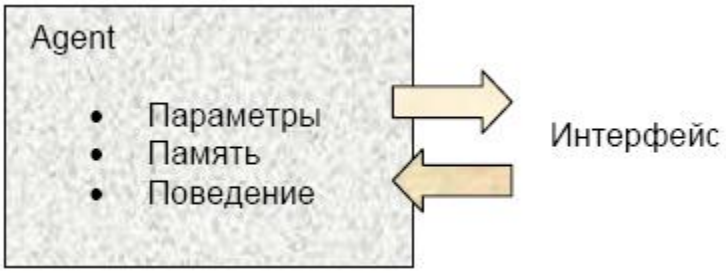


Рис. 17. Активный объект

Параметры могут указывать пол агента, дату рождения и т.д. Переменными можно, например, выразить возраст агента, 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, который реализуется с помощью интерфейсных объектов: портов и интерфейсных переменных.

Работа в AnyLogic. Создайте проект для своей модели и сохраните его в папке. Первым шагом при создании агентной модели является создание агентов. Для каждого агента задается набор правил, согласно которым он взаимодействует с другими агентами; это взаимодействие и определяет

общее поведение системы. Для примера агентами пусть будут люди. Создадим агентную модель с помощью Мастера создания модели. Создание агентной модели.

Шаг 1. Щелкните мышью по кнопке панели инструментов Создать. Появится - Agent • Параметры • Память • Поведение Интерфейс. Появится диалоговое окно Новая модель. Задайте имя новой модели. Щелкните мышью по кнопке Далее.

Шаг 2. Выберите шаблон модели (рис. 18). В связи с тем, что создаем новую агентную модель, то нужно установить флажок Использовать шаблон модели и выбрать Агентная модель в расположенном ниже списке Выберите метод моделирования. Щелкните мышью по кнопке Далее.

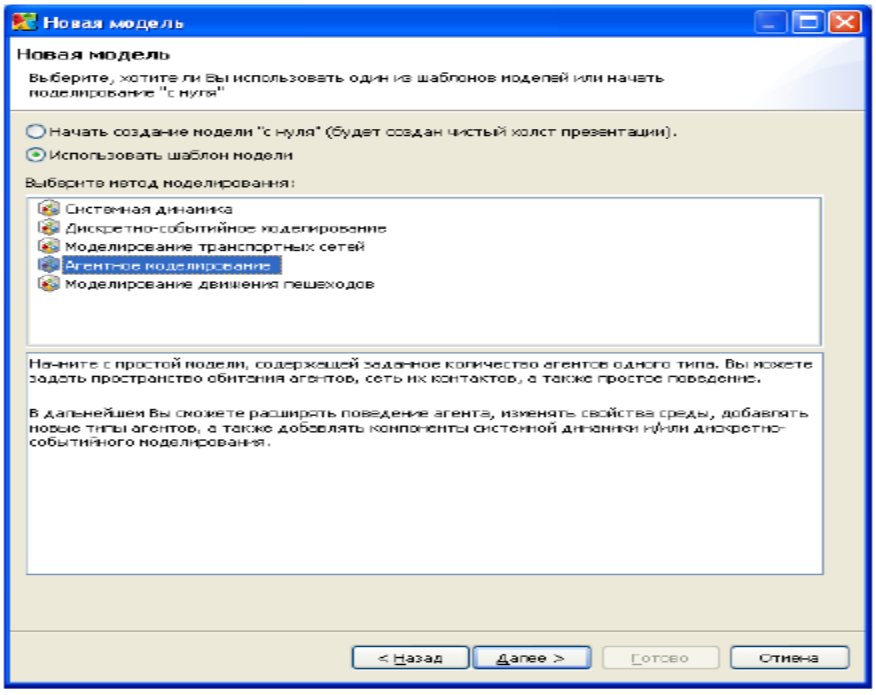


Рис. 18. Агентная модель

Шаг 3. Создайте агентов, т.е. задайте имя класса агента и количество агентов, которое будет изначально создано в нашей модели. Задайте в качестве имени класса Person и введите в поле Начальное количество агентов 1000. Щелкните мышью по кнопке Далее.

Шаг 4. Задайте свойства пространства, в котором будут обитать агенты и выберите фигуру анимации агента. Установите флажок Добавить пространство и выберите ниже тип этого пространства – Непрерывное.

Задайте размерности данного пространства: введите в поле Ширина – 600, а в поле Высота – 350. В результате наши агенты будут располагаться каким-то образом в пределах непрерывного пространства, отображаемого на презентации модели 65 областью размером 600×350 пикселей. Не меняйте значения, выбранные в выпадающих списках Начальное расположение и Анимация;

Пусть агенты изначально расставляются по пространству случайным образом, а анимируются с помощью фигурки человечка. Щелкните мышью по кнопке Далее.

Шаг 5. Задайте сеть взаимосвязей агентов (рис. 19). Установите флажок Использовать сеть и оставьте выбранной опцию Случайное. Ниже можно установите флажок Показывать связи, чтобы отображать на презентации связи между знакомыми (или потенциально могущими встретиться и пообщаться) агентами с помощью линий. Щелкните мышью по кнопке Далее.

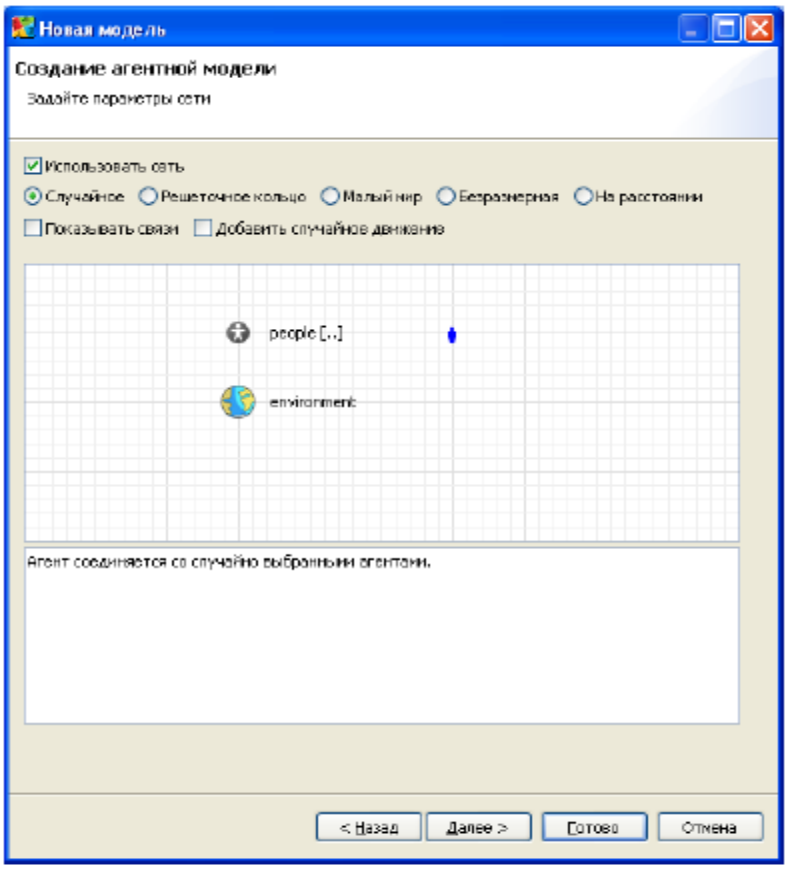


Рис. 19. Сеть взаимосвязей.

Шаг 6. Установите флажок Добавить простое поведение. В результате у агента будет создана диаграмма состояний.

Задание характеристик агента Характеристики агента задаются с помощью параметров класса. Все агенты обладают общей структурой, поскольку все они задаются объектами одного класса. Параметры же позволяют задавать характеристики индивидуально для каждого агента. Создадим параметр, который задает подверженность человека влиянию рекламы.

Откройте структурную диаграмму класса Person. Перетащите элемент Параметр из палитры Основная на диаграмму класса, в окне свойств параметра задайте имя AdEffectiveness, значение по умолчанию – 0.011.

Задание поведения агента Поведение агента обычно описывается в классе этого агента (в этой модели - класс Person) с помощью диаграммы состояний (стейтчарт). Мастер создания моделей уже создал простейшую диаграмму состояний из двух состояний, между которыми существует два разнонаправленных перехода. Изменим данный стейтчарт.

1. Откройте структурную диаграмму класса Person. На диаграмме класса вы увидите следующую диаграмму состояний (рис. 20).

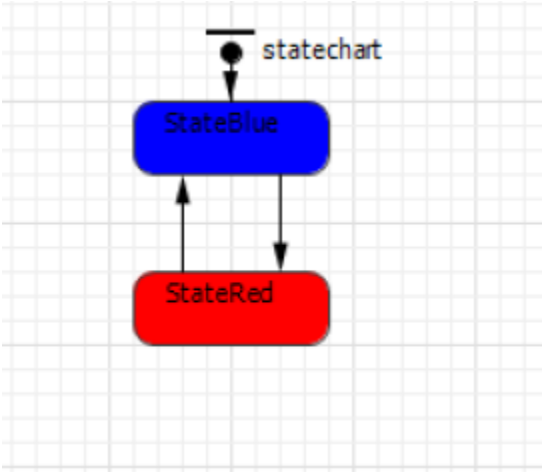


Рис. 20. Диаграмма состояний

2. Откройте свойства верхнего состояния, переименуйте верхнее состояние в PotentialAdopter. Это начальное состояние. Нахождение стейтчарта в данном состоянии означает, что человек еще не купил продукт.

3. Нижнее состояние назовите Adopter (т.е. человек уже купил продукт).

4. Измените свойства перехода из состояния PotentialAdopter в состояние Adopter. Этот переход 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.

5. Удалите переход, ведущий из нижнего состояния в верхнее, поскольку мы пока создаем простейшую модель, в которой человек, однажды приобредший продукт, навсегда остается его потребителем, и соответственно перехода из состояния Adopter в состояние PotentialAdopter пока что быть не должно (рис. 5). Чтобы удалить переход, выделите его на диаграмме и нажмите Del.

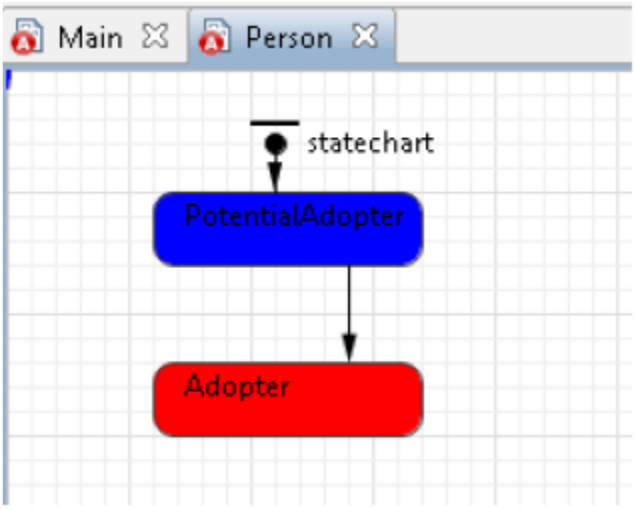


Рис. 21

6. Настройте выполнение модели (рис. 22). В окне свойств эксперимента перейдите на вкладку Модельное время и задайте останов модели после 8 единиц модельного времени.

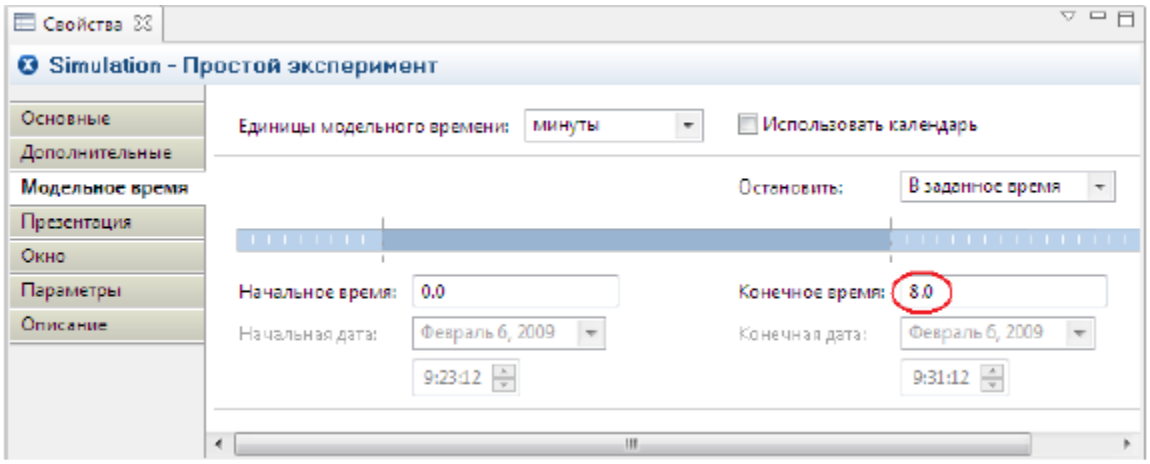


Рис. 22. Нстройка выполнения моделей

7. Постройте проект с помощью кнопки панели инструментов Построить (клавиша F7). Если ошибок в проекте нет, то запустите модель. Вы увидите, как число потенциальных покупателей (синих) переходит в разряд покупателей (красных). 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. Создадим функции сбора статистики для подсчета потенциальных потребителей продукта.

1. Откройте диаграмму класса Main. Выделите на диаграмме вложенный объект people.

2. Перейдите на вкладку Статистика панели свойств объекта people. Щелкните мышью по кнопке Добавить функцию сбора статистики. Откроет- ся секция свойств для задания свойств новой функции сбора статистики по элементам этого реплицированного объекта (people).

3. Задайте имя функции – potentialAdopters. Оставьте выбранный по умолчанию Тип функции – кол-во. Задайте Условие: item.statechart.isStateActive(item.PotentialAdopter) Эта функция будет вести подсчет количества агентов, для которых выполняется заданное условие,

т.е. тех агентов, которые находятся в текущий момент времени в состоянии PotentialAdopter (являются потенциальными потребителями продукта). Здесь item – это агент (элемент реплицированного объекта people).

4. Создайте еще одну функцию сбора статистики (рис. 23). Назовите ее adopters. Тип функции – кол-во. Условие: item.statechart.isStateActive(item.Adopter) Данная функция будет вести подсчет количества агентов, которые находятся в состоянии Adopter (т.е. уже приобрели продукт).

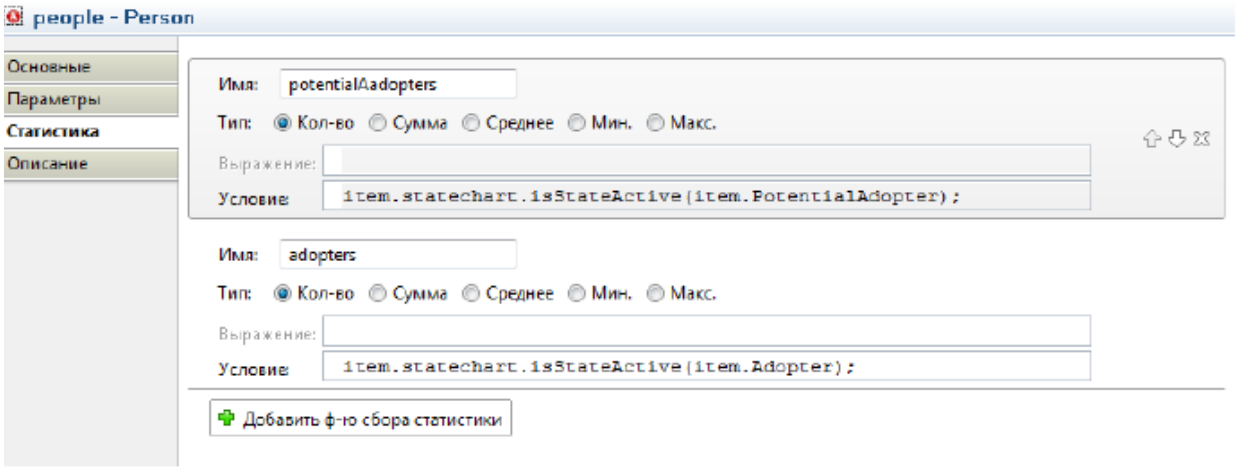


Рис. 23.

Добавьте временной график, отображающий динамику изменения численностей потребителей и потенциальных потребителей продукта. Расположите его, как показано на рис. 24.

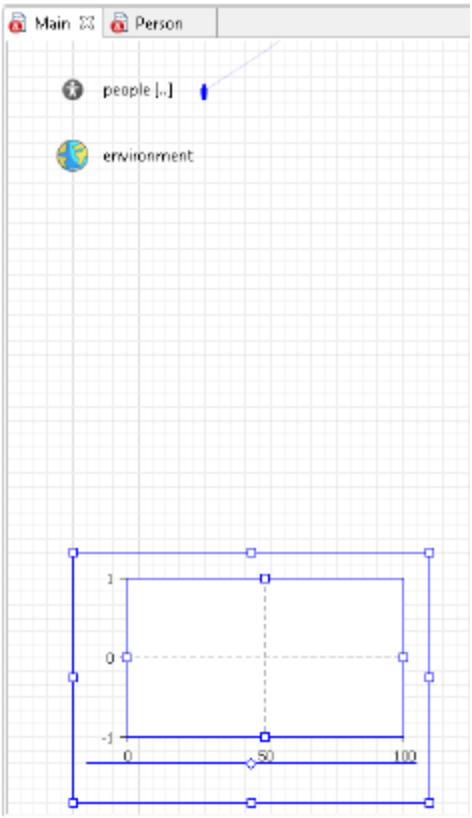


Рис. 24

Настройте свойства графика (рис. 25).

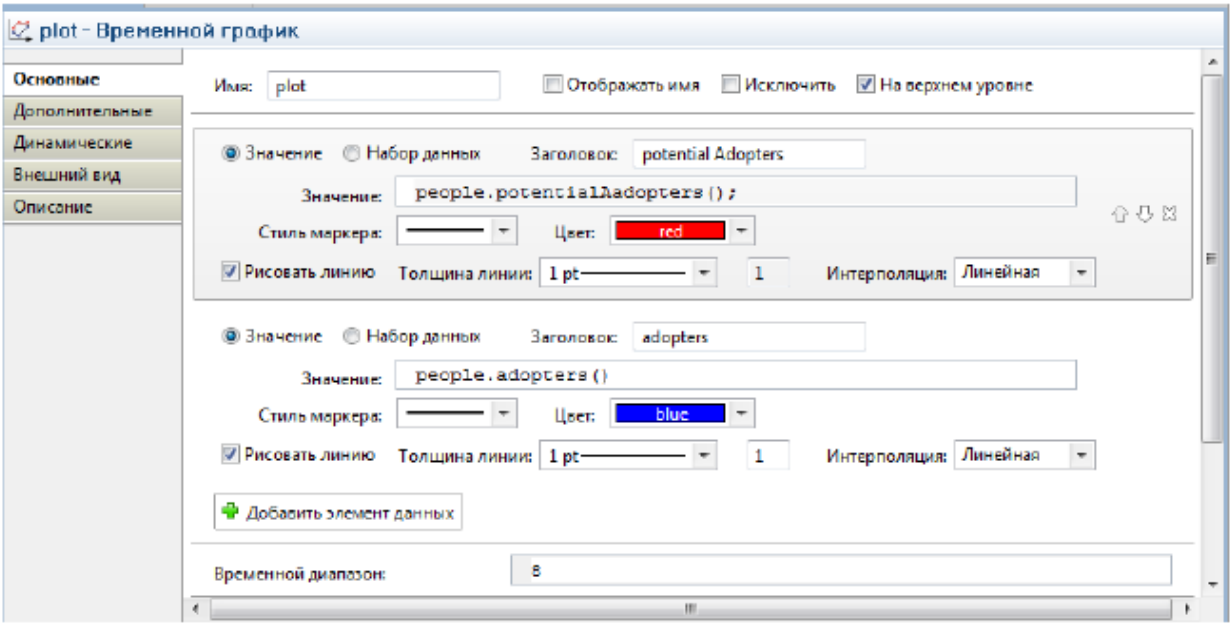


Рис. 25.

Запустите модель. На графике (рис.26) 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.

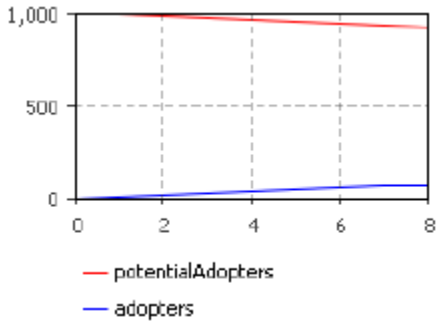


Рис. 26

Результаты работы Студент должен предоставить отчет по лабораторной работе с выводами, продемонстрировать работу модели, ответить на вопросы преподавателя.

Контрольные вопросы

1. Дайте понятие агента и мультиагентной системы
2. Каким образом первоначально агенты располагаются?
3. В чем принципиальные особенности пакета AnyLogic?
4. Что такое палитра в пакете AnyLogic?
5. Для какой цели задается модельное время?

5. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. AnyLogic User’s Manual. XJ Technologies : [электрон. ресурс]. Режим доступа : http://www.xjtek.com

2. AnyLogic Tutorial. XJ Technologies : [электрон. ресурс]. Режим доступа : http://www.xjtek.com