



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**
Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)



УТВЕРЖДАЮ
Директор
Е.А. Дрофа
20.04 2022 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ по
дисциплинам
для студентов направления подготовки
09.04.02 Информационные системы и технологии
программа магистратуры «Информационные системы и технологии»



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

УТВЕРЖДАЮ

Директор

_____ Е.А. Дрофа

_____ 2022 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ по
дисциплинам
для студентов направления подготовки
09.04.02 Информационные системы и технологии
программа магистратуры «Информационные системы и технологии»



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Современные проблемы науки и производства
(информационные системы)» для студентов направления
подготовки

09.04.02 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Современные проблемы науки и производства (информационные системы)» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии, направленность (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1

Лабораторная работа 2

Лабораторная работа 3

Лабораторная работа 4 .

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины – освоение методов разработки математических моделей информационных процессов и методологии и технологии математического моделирования при исследовании, проектировании, эксплуатации информационных систем; формирование общекультурных и профессиональных компетенций магистра в соответствии с требованиями ФГОС по направлению Информационные системы и технологии; подготовка магистра к деятельности, требующей применение научно-практических знаний и умений в области анализа информационных процессов; развитие логического, алгоритмического мышления студентов, умения самостоятельно расширять свои знания в области математического представления информационных процессов.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

В результате освоения данной дисциплины формируется следующая компетенция у обучающегося:

УК-6.2: Решает задачи собственного личностного и профессионального развития, определяет и реализовывает приоритеты совершенствования собственной деятельности; применяет методики, позволяющие улучшить и сохранить здоровье в процессе жизнедеятельности

УК-5.1: Формулирует закономерности и особенности социально-исторического развития различных культур; учитывает особенности межкультурного разнообразия общества

УК-5.2: Понимает и толерантно воспринимает межкультурное разнообразие общества; анализирует и учитывает разнообразие культур в процессе межкультурного взаимодействия

Изучив данный курс, студент должен:

Знать:

информационные и телекоммуникационные технологии в науке и образовании; современные тенденции в проведении исследований по информационным системам;

современные мировые тенденции в разработке новых информационных систем и технологий;

основные тенденции в области эффективного использования информационных ресурсов в науке, образовании и промышленности; энергосберегающие технологии, используемые в центрах обработки данных.

Уметь:

формулировать и решать задачи, возникающие в производственной и научно-исследовательской сфере для различных парадигм построения информационных систем;

использовать современные методы, средства и технологии при разработке информационных систем;

осуществлять сбор, обработку, анализ и систематизацию научно-технической информации.

Владеть:

взаимодействия с коллегами и коллективом при исследовании и разработке информационных систем.

Реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа 1 Исследование модели непрерывно-детерминированной системы

Целью занятия является освоение методов моделирования непрерывно-детерминированных динамических систем. В процессе выполнения задания студенты должны:

разработать модели заданной системы, реализованные в пакете визуального моделирования Simulink на основе полученного описания;

получить результаты исследования данных моделей и выполнить их объяснение в терминах прикладной области.

В качестве примера учебного задания рассматривается задача об управлении протеканием воды через систему, состоящую из двух цилиндрических баков, расположенных на разной высоте, или, более кратко, задача о двух баках. Выбор этой задачи в качестве примера обусловлен следующими причинами:

понятным физическим принципом;

нелинейным поведением компонент системы;

наличием аварийных ситуаций в поведении системы, которые необходимо

обрабатывать;

Модель системы, реализованная в подсистеме Simulink пакета Matlab

Построение модели в Simulink сводится к перемещению с помощью мыши необходимых блоков из библиотек Simulink в окно создаваемой модели и соединению этих блоков между собой с помощью функциональных связей. Библиотеки Simulink содержат большое количество разнообразных функциональных блоков, которые отображаются на экране в виде пиктограмм.

Используя описание задачи о двух баках можно выделить в данном примере подсистему System (рис.3), представляющую собой составной блок, который содержит в себе функциональную схему, содержащую в себе диаграмму Stateflow (представленную блоком Controller, являющимся экземпляром стандартного блока Chart и описывающую поведение контроллера) и составной блок Tank_System_Block, соединенные соответствующими функциональными связями (рис.4). Так же в блоке System присутствуют часы Clock, подающие системное время из Simulink в диаграмму Stateflow. Это объясняется тем, что при построении моделей, где используются блоки Stateflow (содержащие в себе переходы, инициируемые истечением неких временных интервалов) необходимо синхронизировать внутреннее системное время в Stateflow и в Simulink. Для этого необходимо на отдельный вход в блоке Stateflow подавать системное время из Simulink и именно это время использовать при составлении условий переходов (таких как истечение отрезков времени Time1 и Time2 в задаче о двух баках). Как видно из рис.4, к связи, соединяющей блок Clock и Controller, подсоединены два экземпляра блока Hit Cross. Еще два экземпляра блока Hit Cross подсоединены к связи блока Controller и выхода h2 блока Tank_System_Block. Использование Hit Cross блоков необходимо для правильного выполнения переходов в диаграмме Stateflow, включенной в модель

Simulink, в которой происходит непрерывное интегрирование. На вход Hit Cross блока подается некая величина (в модели двух баков для двух Hit Cross блоков это системное время, для еще двух - величина h2). Сам Hit Cross блок содержит в себе некую величину, при совпадении которой с величиной, подаваемой на вход Hit Cross блока, система уменьшает шаг интегрирования. В связи со спецификой системы Stateflow это влияет на правильное выполнение переходов и не позволяет системе «проскочить момент», когда должен выполняться тот или иной переход. У первого Hit Cross блока внутренней величиной является момент времени Time1, у второго - Time 1+Time2, у третьего - значение L_plus, у четвертого - L_minus.

Блок System не имеет входов и имеет два выхода - h1 и h2, соединенные со стандартным блоком Mux (объединяющим их в вектор (h1,h2)), соединенный со стандартным блоком вывода Scope (на который он подает вектор выходных величин (h1,h2)).

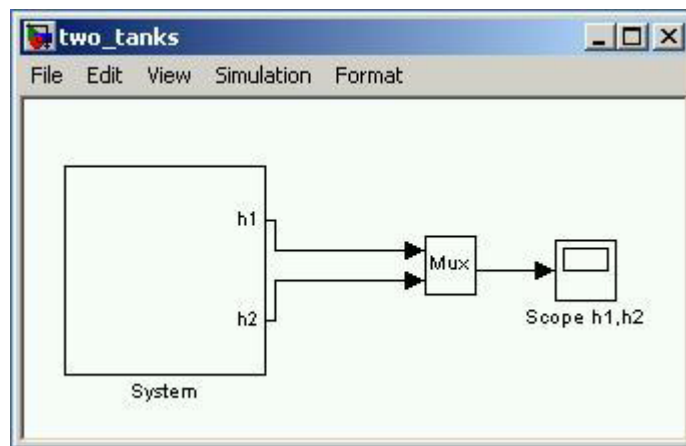


Рисунок 3

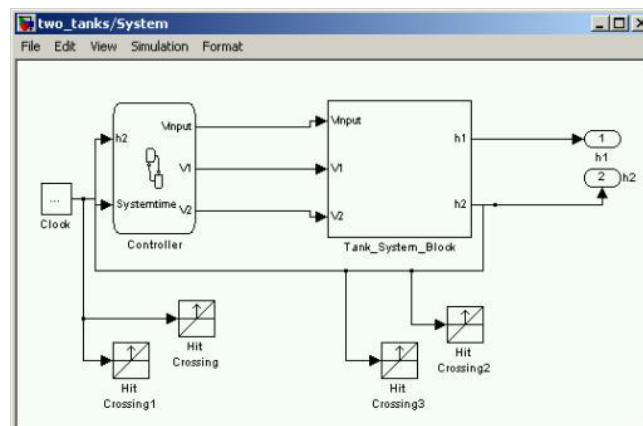


Рисунок 4

В свою очередь, составной блок Tank_System_Block содержит в себе функциональную схему, состоящую из составного блока Tank_System (имеющего три входа, соединенные с соответствующими выходами блока Chart, на которые подаются сигналы контроллера, и два выхода - h1 и h2) и пар экземпляров стандартных блоков Const, Relational Operator и Stop Simulation (выполняющих проверку на переполнение баков), соединенных соответствующими функциональными связями (рис.5). Блок Tank_System_Block имеет три входа (на которые подаются сигналы от контроллера) и два выхода (h1 и h2).

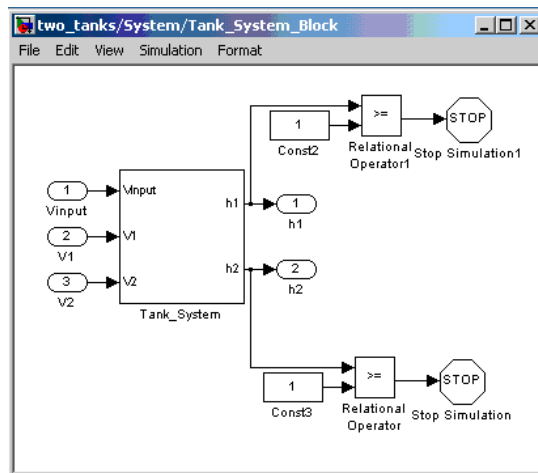


Рисунок 5

Составной блок Tank_System содержит в себе функциональную схему, состоящую из составных блоков Two_Tanks, Vin_Control, K1_Control и K2_Control, соединенных между собой функциональными связями (рис.6). Блок Tank_System имеет то же количество входов и выходов с теми же значениями, что и предыдущий блок.

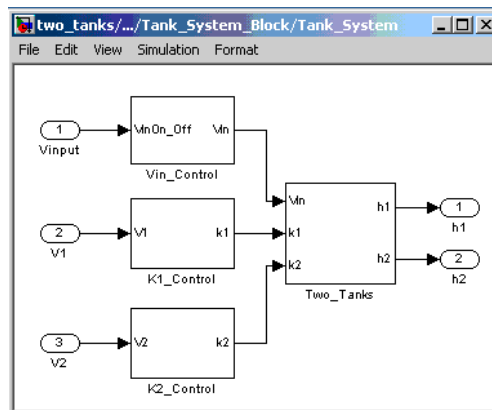


Рисунок 6

Составной блок Vin_Control содержит в себе функциональную схему, состоящую из стандартного блока Switch и двух экземпляров стандартного блока Const, содержащих в себе значения, соответствующие скорости входного потока воды в систему двух баков, когда входной кран открыт и закрыт. Переключения между этими значениями происходит в блоке Switch в зависимости от значения, поступающего на вход блока Vin_Control от контроллера. Соответствующее значение Vin подается на выход. Блок Vin_Control имеет один вход и один выход (рис.7):

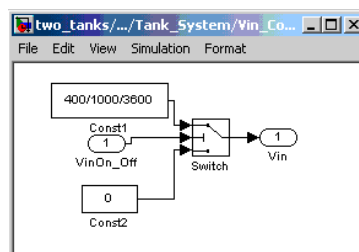


Рисунок 7

Составные блоки K1_Control и K2_Control содержат в себе идентичные функциональные схемы, различающиеся только значением коэффициентов в уравнении, представленном в стандартном блоке fcn. Функциональная схема состоит из двух экземпляров стандартного блока Switch (один из которых необходим для переключения между положениями кранов V1/V2 открыто/закрыто, а второй для отслеживания ситуации, когда параметр p становится равным 80 и соответствующего переключения между значениями функции K(p)), трех экземпляров стандартного блока Const (содержащих в себе значения, 1 и -1, соответствующие положению крана V1/V2 (открывается/закрывается) и значение 0 для функции K(p) в ситуации, когда $p \geq 80$), стандартного блока Integrator (интегрирующего поступающее от переключателя значение в пределах от 0 до 80 с начальным значением интегрируемой величины 80) и стандартного блока f(u), в котором происходит вычисление значения функции K1(p)/K2(p). Переключения в блоке Switch происходит в зависимости от значения, поступающего на вход блока K1_Control/K2_Control от контроллера. Соответствующее значение k1/k2 подается на выход. Блок K1_Control/K2_Control имеет один вход и один выход (на рис.8 представлен блок K1_Control, блок K2_Control идентичен):

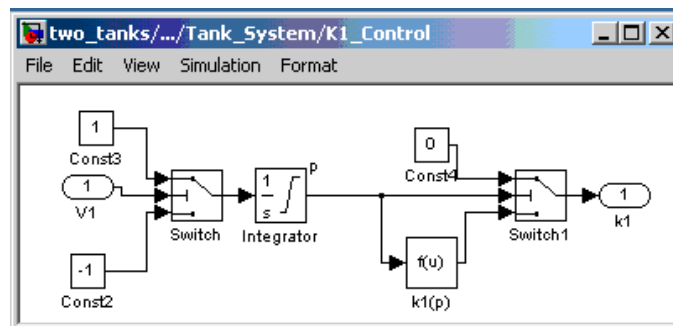


Рисунок 8

Составной блок Two_Tanks содержит в себе функциональную схему, состоящую из составных блоков Vout_calc, V12_calc1 и V12_calc2 (отвечающих за вычисление значений Vout и V12), стандартного блока Switch, двух экземпляров стандартного блока Mux (объединяющих в вектора соответствующие значения - Vin (поступающее на вход блока Two_Tanks с выхода блока Vin_Control)) и V12, а также V12 и Vout), двух экземпляров стандартного блока fcn (производящих вычисление производных h1 и h2), двух экземпляров стандартного блока Integrator, выходами которых являются значения h1 и h2, поступающие на выход. Переключения в блоке Switch происходит в зависимости от значения h2, поступающего на вход блока Switch от интегратора, и в зависимости от этого значения, подключается один из блоков, вычисляющих V12. Блок Two_Tanks имеет три входа, на которые подаются значения Vin, k1 и k2; и два выхода - h1 и h2 (рис.9):

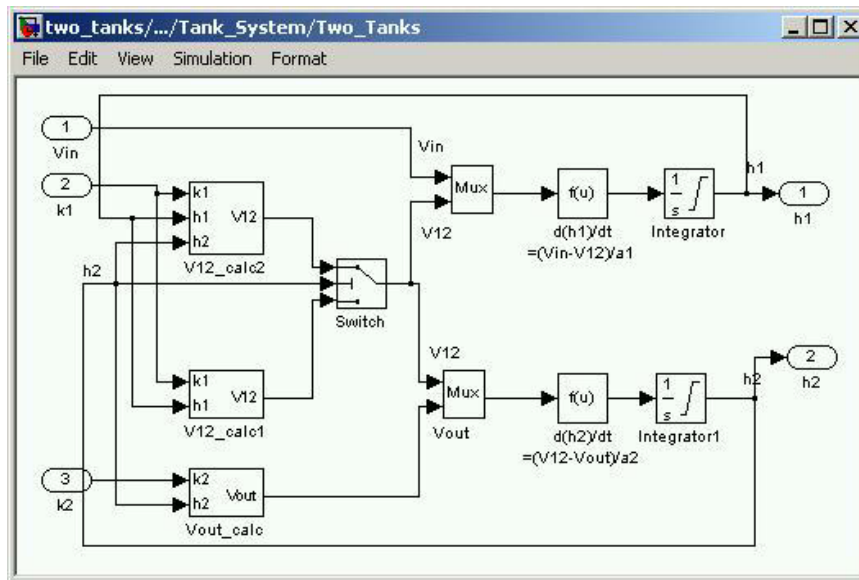


Рисунок 9

Составной блок Vout_calc содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k2 и h2, поступающие на вход блока Vout_calc, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины Vout. Блок Vout_calc имеет два входа и один выход (рис.10):

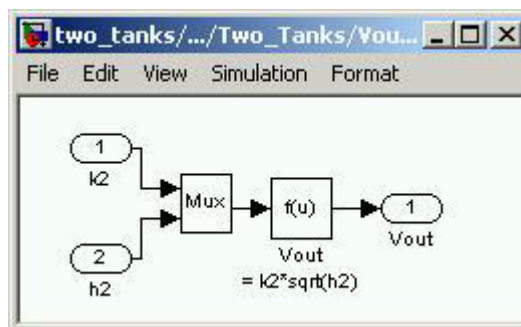


Рисунок 10

Составной блок V12_calc1 содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k1 и h1, поступающие на вход блока V12_calc1, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины V12. Блок V12_calc1 имеет два входа и один выход (рис.11):

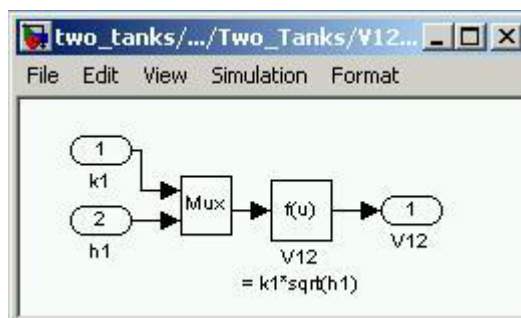


Рис. 11

Составной блок V12_calc2 содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k1, h1 и h2, поступающие на вход блока V12_calc2, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины V12. Блок V12_calc2 имеет три входа и один выход (рис.12):

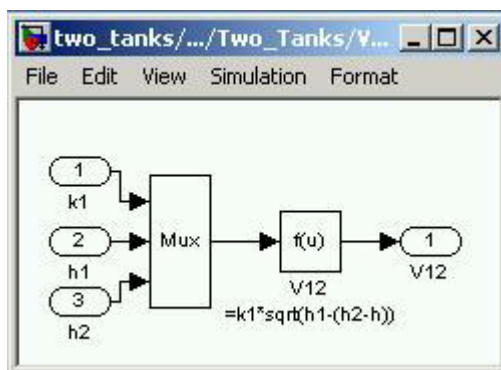


Рис. 12

Диаграмма Stateflow, представлена экземпляром стандартного блока Chart, (рис.13). В ней вместо внутреннего времени t используется подаваемое на вход блока системное время Simulink (входная переменная Systemtime).

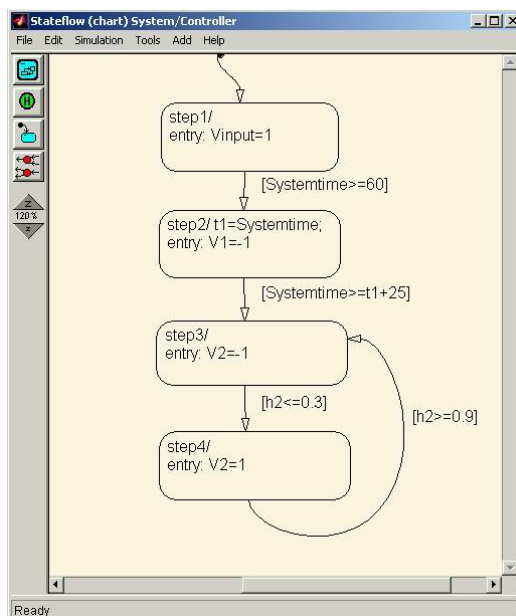


Рис. 13

Результаты эксперимента

На рис.14 представлена временная диаграмма изменений уровней воды в баках, полученная в результате эксперимента с моделью, реализованной в подсистеме Simulink пакета Matlab:

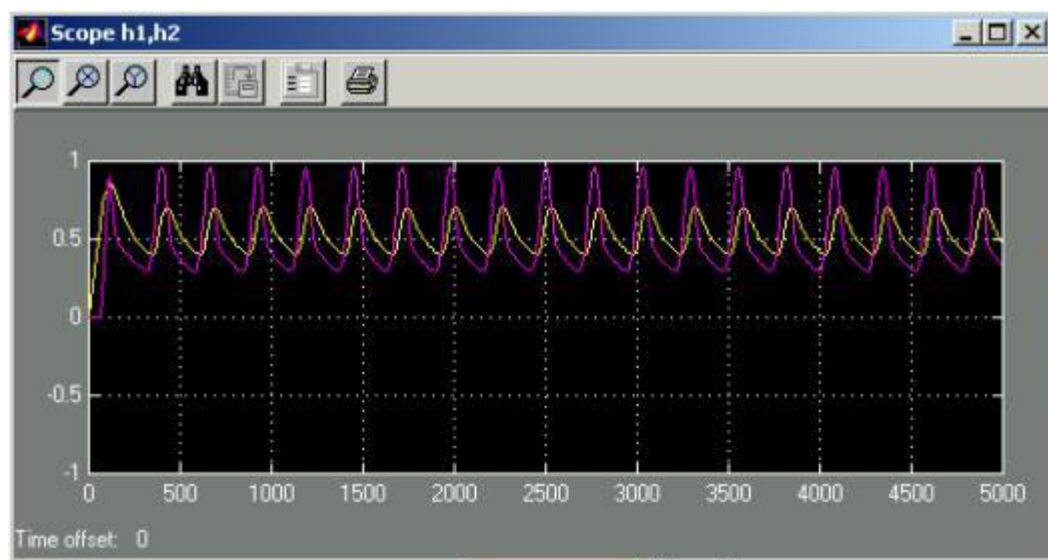


Рис. 14

Данные временные диаграммы отражают изменения уровней воды в системе двух баков при открытии/закрытии выходного крана. Возрастающие участки кривых $h_1(t)$ и $h_2(t)$ соответствуют состоянию системы, когда выходной кран закрыт. Убывающие участки кривых $h_1(t)$ и $h_2(t)$ соответствуют состоянию системы, когда выходной кран открыт.

Лабораторная работа №2 Исследование стохастической имитационной модели системы

Целью занятия является практическое освоение методов моделирования стохастических динамических систем. В процессе выполнения задания студенты должны:

разработать модель заданной системы с использованием пакета визуального моделирования Simulink на основе полученного описания;

получить результаты исследования модели и выполнить их объяснение в терминах прикладной области.

1. Описание моделируемой системы

В практике передачи и приема информации с использованием радиоканалов связи одной из важнейших задач является оценка влияния помех на вероятность безошибочной передачи информации. Рассмотрим систему цифрового радиоканала, представленную на рис. 1.

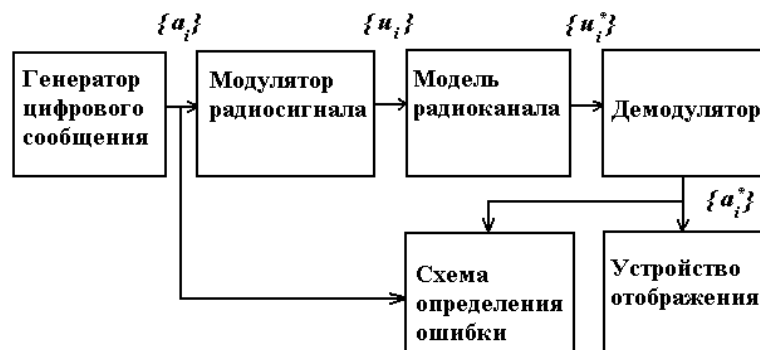


Рисунок 1

Генератор цифровых сообщений служит для имитации двоичных кодов заданной длины и может быть представлен следующим выражением

$$\begin{cases} a_i = 0, \text{ если } Rnd(i) < 0,5 \\ a_i = 1, \text{ если } Rnd(i) \geq 0,5 \end{cases}$$

где $Rnd(i)$ число из интервала $0 \dots 1$, выдаваемое встроенным программным генератором псевдослучайных чисел; $i = \overline{1, I_{зад}}$.

Несущий радиосигнал задан выражением

$$u(t) = U \sin(2\pi f_n t + \varphi_0),$$

где U амплитуда радиосигнала; f_n - несущая частота; φ_0 - начальная фаза.

Модуляция радиосигнала по закону передаваемого сообщения может быть выполнена с помощью импульсной модуляции (ИМ):

$$\begin{cases} U = 0, \text{ если } a_i = 0; \\ U = 1, B, \text{ если } a_i = 1. \end{cases}$$

Частотной модуляции (ЧМ):

$$\begin{cases} f = f_n, \text{ если } a_i = 0; \\ f = f_n + \Delta f, \text{ если } a_i = 1. \end{cases}$$

Фазокодовой манипуляции (ФКМ):

$$\begin{cases} \varphi_0 = 0, \text{ если } a_i = 0; \\ \varphi_0 = \pi, \text{ если } a_i = 1. \end{cases}$$

Модель канала характеризуется видом помех. Для аддитивных помех, приводящих к искажению амплитуды сигнала, используется выражение

$$u^*(t) = u(t) \pm Rnd(t) \cdot U_{\Pi},$$

где U_{Π} - максимальное значение амплитуды помехи.

Модель канала с фазовыми искажениями имеет вид

$$u^*(t) = u(t) + U_{\Pi} \cdot \sin(2\pi f_n t \pm Rnd(t) \cdot \varphi_0).$$

Модель канала с преобладающими частотными искажениями может быть задана выражением

$$u^*(t) = u(t) + U_{\Pi} \cdot \sin(2\pi t(f_n \pm Rnd(t) \cdot \Delta f)).$$

Демодуляция определяется видом модуляции радиосигнала и для амплитудной импульсной модуляции задается выражением

$$\begin{cases} u^*(t) > U_{порога}^1, \text{ то } a_i^* = 1; \\ u^*(t) < U_{порога}^0, \text{ то } a_i^* = 0, \text{ иначе } a_i^* = x. \end{cases}$$

Пороговые значения нулевого и единичного уровней определяются исходя из условий распространения сигнала и дальности связи. Состояние x характеризует появление ошибки. Аналогичные выражения только с пороговыми значениями частоты или фазы сигнала используются для демодуляции ЧМ и ФКМ сигналов.

Фиксация ошибки осуществляется посредством сравнения $a_i = a_i^*$ или по наличию ошибки демодуляции $a_i^* = x$.

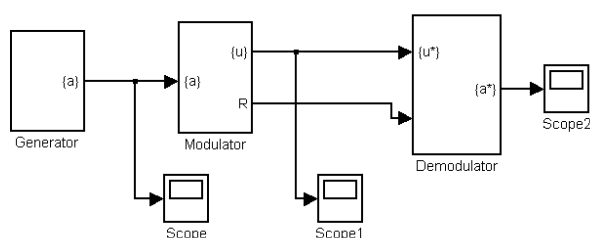
2. Задание для самостоятельной работы

В соответствии с номером по журнальному списку выбрать из табл. 1 исходные данные для моделирования и составить модель радиоканала. При составлении модели максимально использовать возможности пакета имитационного моделирования Simulink. Оформить отчет по практическому занятию в котором отразить текст описания системы и разработанной модели. Привести графики на выходе элементов модели для различных вариантов исходных данных. Оценить влияние различных помеховых воздействий на число ошибок.

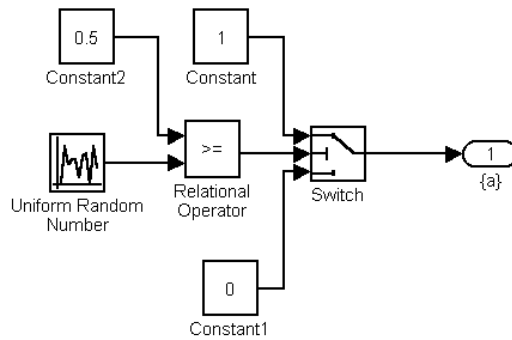
Таблица 1

№ п/п	Исходные данные для моделирования					Примечание
	Вид модуляции	Число разрядов	$U^1_{порога}$	$U^0_{порога}$	$U_{П}$	
1	ИМ	7	$0,6U$	$0,1U$	$0,01U$	
2	ФКМ	8				
3	ЧМ	9				$\Delta f = 0,1 f$
4	ИМ	10	$0,8U$	$0,2U$	$0,02U$	
5	ЧМ	7				$\Delta f = 0,2 f$
6	ФКМ	8				
7	ЧМ	9				$\Delta f = 0,01 f$
8	ИМ	10	$0,9U$	$0,1U$	$0,01U$	
9	ФКМ	7				
10	ИМ	8	$0,5U$	$0,05U$	$0,01U$	

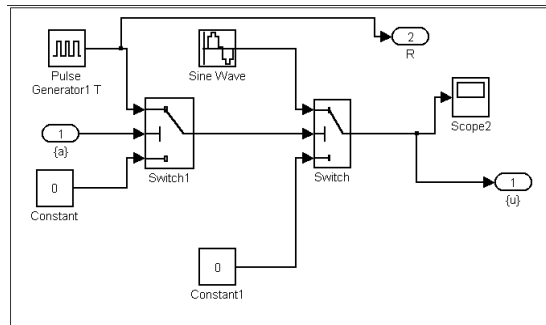
Структура модели



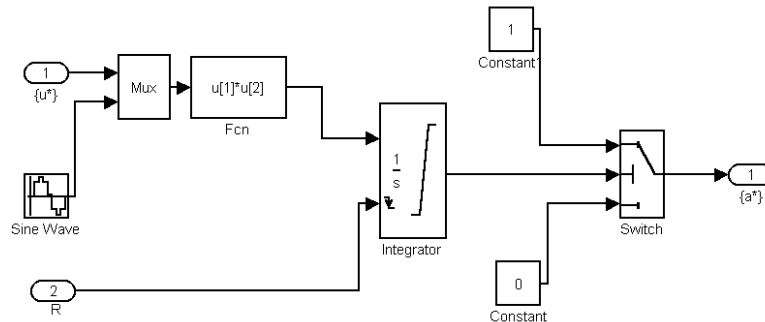
Модель генератора цифрового сигнала



Модель модулятора радиосигнала



Модель демодулятора радиосигнала



Исходные данные моделей

Время моделирования 1000.

Генератор случайных чисел:

Минимальное значение случайного числа =0; максимальное значение =1; начальное значение последовательности (любое целое число); Длительность одной реализации =100.

Minimum=0; Maximum=1; Initial seed=0; Sample time=100.

Генератор импульсов:

Период =100 секунд модели; Длительность импульса =50 (скважность равна 2); амплитуда 1 вольт; начальное время генерации =0.

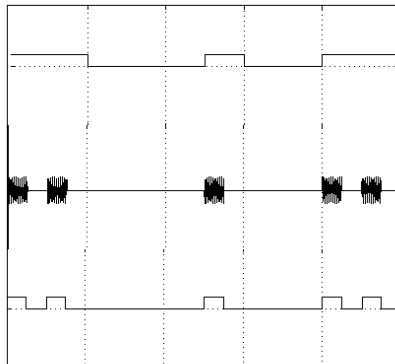
Period=100(secs); Duty cycle (% of period)=50; Amplitude=1; Start time=0.

Генератор синусоиды:

Амплитуда 1 вольт; Циклическая частота=10 радиан в секунду; Начальная фаза =0; Длительность одного значения 1.

Amplitude=1; Frequency (rad/sec)=10; Faze (rad)=0; Sample time=1.

Пример результатов моделирования



Лабораторная работа №3

Исследование элементов модели системы массового обслуживания

Цель занятия: Освоить методику моделирования элементов систем массового обслуживания (СМО)

1. Общие сведения о СМО

Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования.

В любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания заявкой и собственно обслуживание заявки. Это можно изобразить в виде некоторого i -го прибора обслуживания Π_i (рис. 1), состоящего из накопителя заявок H_i в котором может одновременно находиться $l_i = \overline{L_i^H}$ заявок, где L_i^H — емкость i -го накопителя, и канала обслуживания заявок (или просто канала) K_i . На каждый элемент прибора обслуживания Π_i поступают потоки событий: в накопитель H_i — поток заявок w_i на канал K_i — поток обслуживания u_i .

Потоком событий называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Различают потоки однородных и неоднородных событий. Поток событий называется *однородным*, если он характеризуется только моментами поступления этих событий (вызывающими моментами) и задается последовательностью $\{t_n\} = \{0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq \dots\}$, где t_n — момент наступления n -го события — неотрицательное вещественное число. Однородный поток событий также может быть задан в виде последовательности промежутков времени между n -м и $(n-1)$ -м событиями $\{\tau_n\}$, которая однозначно связана с последовательностью вызывающих моментов $\tau_n = t_n - t_{n-1}, n \geq 1, t_0 = 0$, т. е. $\tau_1 = t_1$.

Потоком неоднородных событий называется последовательность $\{t_n, f_n\}$, где t_n — вызывающие моменты; f_n — набор признаков события. Например, применительно к процессу обслуживания для неоднородного потока заявок могут быть заданы принадлежность к тому или иному источнику заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т. п.

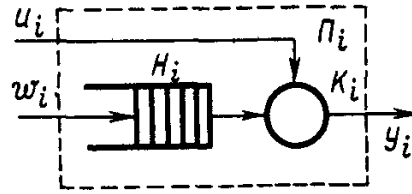


Рис. 1. Устройство обслуживания заявок

Рассмотрим поток, в котором события разделены интервалами времени τ_1, τ_2, \dots , которые вообще являются случайными величинами. Пусть интервалы τ_1, τ_2, \dots независимы между собой. Тогда поток событий называется *поток с ограниченным последствием*.

Пример потока событий приведен на рис. 2, где обозначено T_j интервал между событиями (случайная величина); T_n — время наблюдения, T_c — момент совершения события.

Интенсивность потока можно рассчитать экспериментально по формуле

$$\lambda = \frac{N}{T_n}$$

где N — число событий, произошедших за время наблюдения T_n . Если $T_j = \text{const}$ или определено какой-либо формулой $T_j = f(T_{j-1})$, то поток называется *детерминированным*. Иначе поток называется *случайным*.

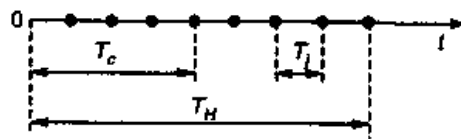


Рис.2 Графическое изображение N -схемы

Случайные потоки бывают:

- ординарными, когда вероятность одновременного появления 2-х и более событий равна нулю;
- стационарными, когда частота появления событий постоянная;
- без последствия, когда вероятность не зависит от момента предыдущих событий.

Поток событий называется *ординарным*, если вероятность того, что на интервал времени, примыкающий к моменту времени t , попадает больше одного события, пренебрежительно мала по сравнению с вероятностью того, что на этот же интервал времени попадает ровно одно событие.

Стационарным потоком событий называется поток, для которого вероятность появления того или иного числа событий на интервале времени t зависит от длины этого участка и не зависит от того, где на оси времени взят этот участок.

Интенсивность потока может быть любой неотрицательной функцией времени, имеющей размерность, обратную размерности времени. Для стационарного потока его интенсивность не зависит от времени и представляет собой постоянное значение, равное среднему числу событий, наступающих в единицу времени.

Возможные приложения. Обычно в приложениях при моделировании различных систем применительно к элементарному каналу обслуживания K_j , можно считать, что поток заявок $W_j \in W$, т. е. интервалы времени между моментами появления заявок (вызывающие моменты) на входе K_j образует подмножество неуправляемых переменных, а поток обслуживания $u_j \in U$, т. е. интервалы времени между началом и окончанием обслуживания заявки, образует подмножество управляемых переменных.

Заявки, обслуженные каналом K_j и заявки, покинувшие прибор Π_j по различным причинам необслуженными (например, из-за переполнения накопителя H_j , образуют выходной поток $y_j \in Y$, т. е. интервалы времени между моментами выхода заявок образуют подмножество выходных переменных.

Процесс функционирования прибора обслуживания Π_j можно представить как процесс изменения состояний его элементов во времени $z_j(t)$. Переход в новое состояние для Π_j означает изменение количества заявок, которые в нем находятся (в канале K_j , и в накопителе H_j). Таким образом, вектор состояний для Π_j имеет вид $\vec{z}(t) = (z_j^K, z_j^H)$, где z_j^H — состояние накопителя ($z_j^H = 0$ — накопитель пуст, $z_j^H = 1$ — в накопителе имеется одна заявка ... $z_j^H = L_j^H$ — накопитель полностью заполнен); L_j^H — емкость накопителя, измеряемая числом заявок, которые в нем могут поместиться; z_j^K — состояние канала ($z_j^K = 0$ — канал свободен, $z_j^K = 1$ — канал занят и т. д.).

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а *Q-схемы*, образуемые композицией многих элементарных приборов обслуживания (сети массового обслуживания). Если каналы K_i различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание (многоканальная *Q-схема*), а если приборы Π_j и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная *Q-схема*). Таким образом, для задания *Q-схемы* необходимо использовать оператор сопряжения R , отражающий взаимосвязь элементов структуры (каналов и накопителей) между собой.

Связи между элементами *Q-схемы* изображают в виде стрелок (линий потока,

отражающих направление движения заявок). Различают разомкнутые и замкнутые Q -схемы. В разомкнутой Q -схеме выходной поток обслуженных заявок не может снова поступить на какой-либо элемент, т. е. обратная связь отсутствует, а в замкнутых Q -схемах имеются обратные связи, по которым заявки двигаются в направлении, обратном движению вход-выход.

Собственными (внутренними) параметрами Q -схемы будут являться количество фаз обслуживания, количество каналов в каждой фазе, количество накопителей каждой фазы, емкость i -го накопителя. Следует отметить, что в теории массового обслуживания в зависимости от емкости накопителя применяют следующую терминологию для систем массового обслуживания: системы с потерями (т. е. накопитель отсутствует, а имеется только канал обслуживания), системы с ожиданием (т. е. накопитель, имеет бесконечную емкость и очередь заявок не ограничивается) и системы смешанного типа (с ограниченной емкостью накопителя). Всю совокупность собственных параметров Q -схемы обозначим как подмножество H ,

Для задания Q -схемы также необходимо описать алгоритмы ее функционирования, которые определяют набор правил поведения заявок в системе в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе и обслуживания заявок каналом, каждого элементарного обслуживающего прибора P , Q -схемы. Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов.

В зависимости от динамики приоритетов в Q -схемах различают статические и динамические приоритеты. Статические приоритета назначаются заранее и не зависят от состояний Q -схемы, т. е. они являются фиксированными в пределах решения конкретной задачи моделирования. Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций. Исходя из правил выбора заявок из накопителя на обслуживание каналом, можно выделить относительные и абсолютные приоритеты. *Относительный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель ожидает окончания обслуживания предшествующей заявки каналом K , и только после этого занимает канал. *Абсолютный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель прерывает обслуживание каналом K заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из K заявка может либо покинуть систему, либо может быть снова записана на какое-то место в H).

При рассмотрении алгоритмов функционирования приборов обслуживания (каналов, и накопителей) необходимо также задать набор правил, по которым заявки покидают P и K . Для P — либо правила переполнения, по которым заявки в зависимости от заполнения H покидают систему, либо правила ухода, связанные с истечением времени ожидания заявки в H , для K — правила выбора маршрутов или направлений ухода. Кроме того, для заявок необходимо задать правила, по которым они остаются в канале K или не допускаются до обслуживания каналом K т. е. правила блокировок канала. При этом различают блокировки K по выходу и по входу. Такие блокировки отражают наличие управляющих связей в Q -схеме, регулирующих поток заявок в зависимости от состояний Q -схемы. Весь набор возможных алгоритмов поведения заявок в Q -схеме можно представить в виде некоторого оператора алгоритмов поведения заявок A .

Таким образом, Q -схема, описывающая процесс функционирования системы массового

обслуживания любой сложности, однозначно задается в виде $Q = \langle W, U, H, Z, R, A \rangle$.

2. Моделирование основных элементов СМО с использованием пакета

Simulink

В состав модели СМО в обязательном порядке входят модели генераторов потока заявок и потока обслуженных заявок. В основу моделей генераторов положены источники последовательностей случайных чисел равномерно распределенных на интервале $(0 \dots 1)$. На основе этой последовательности можно имитировать потоки с различными законами распределения.

Пусть при моделировании некоторой системы необходимо сформировать на ЭВМ простейший поток заявок. Тогда длина интервала между $(i-1)$ -м и i -м событиями $y_i = -(1/\lambda) \ln(x_i)$.

Если при моделировании некоторой системы требуется сформировать на ЭВМ поток событий, равномерно распределенных на интервале (a, b) , то функция плотности интервалов между событиями $f(y) = 1/(b-a)$, $a \leq y \leq b$. Распределение первого интервала между началом отсчета и первым событием

$$f_1(y_1) = \lambda \left(1 - \int_0^{y_1} f(y) dy\right) = \lambda \left[1 - \int_0^{y_1} dy/(b-a)\right].$$

Интенсивность потока

$$\lambda = 1/M[y] = 1/\int_a^b y f(y) dy = 2/(a+b).$$

Тогда $f_1(y_1) = 2[1 - y_1/(b-a)]/(a+b)$.

где x_i — случайная величина, равномерно распределенная на интервале $(0, 1)$.

Для формирования на ЭВМ потока Эрланга, в котором между последовательными событиями закон распределения интервалов

$$f_k(t) = \frac{\lambda (\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, \quad t > 0.$$

плотность распределения длины первого интервала

$$f_1(y_1) = a'(t_0, y_1) e^{-a(t_0, y_1)},$$

где a — математическое ожидание числа событий на интервале $(t_0 + \Delta t)$, $a(t_0, y_1) = -\ln(x_1)$.

Примерный состав модели генератора представлен на рис.2

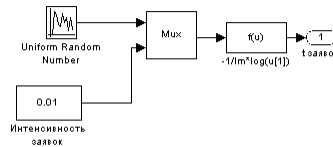


Рис.2 . Генератор

Функция, реализованная в генераторе, соответствует заданному закону распределения случайных величин. Генератор обслуженных заявок имеет аналогичную структуру.

Основной задачей накопителя является сравнение времени обслуживания заявки и времени поступления новой заявки. Если предыдущая заявка не была обслужена, формируется признак наличия задержанной заявки. На рис.3 показан примерный вид накопителя.

Вариант объединения накопителя и двух генераторов показан на рис.4.

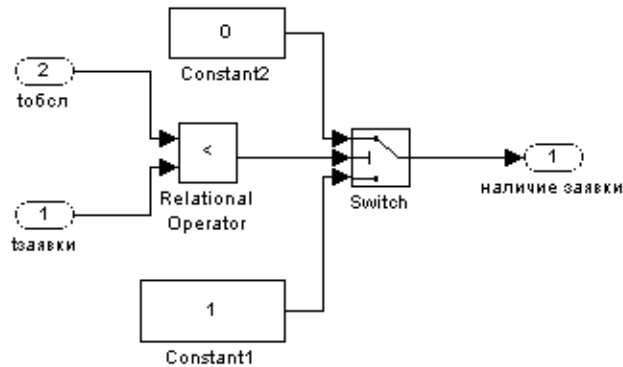


Рис. 3. Накопитель

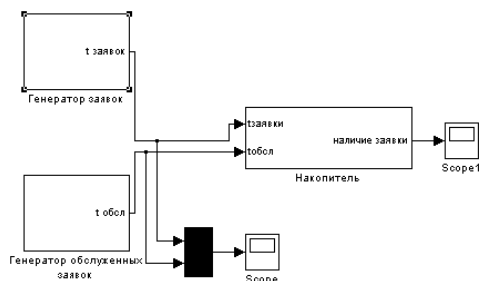


Рис. Пример соединения накопителя и генераторов

Лабораторная работа №4.

Исследование модели системы массового обслуживания

Цель занятия: Освоить методику моделирования элементов систем массового обслуживания (СМО)

В зависимости от характера источника заявок различают разомкнутые и замкнутые системы массового обслуживания СМО. На практическом занятии будут рассмотрены только разомкнутые СМО (рис.1). В зависимости от числа мест в очереди различают СМО с отказами и без отказов. В СМО с отказами число мест в очереди конечно и вследствие вероятностного характера, как входящего потока, так и процессов обслуживания, существует ненулевая вероятность того, что поступившая на вход СМО заявка застанет все каналы занятыми обслуживанием и все места в очереди занятыми заявками, ожидающими обслуживания, т. е. она получит отказ. В СМО без отказов заявка либо сразу назначается на обслуживание, если в момент ее поступления свободен хотя бы один канал, либо безусловно принимается в очередь.

1. Построение сетевых моделей одноканальных систем массового обслуживания

Кратко рассмотрим методику составления сетевых моделей на примере одноканальной СМО (в совокупности с генератором заявок). Следует отметить, что процесс разработки сетевых моделей в общем случае является неформализованным. Проинтерпретируем СМО в терминах транзакций и ресурсов. Транзакции – это активные подвижные элементы системы, а ресурсы – неактивные. Транзакциями в СМО являются заявки, а ресурсом является канал обслуживания заявок. Функционирование СМО описывается как взаимодействие транзакций и ресурсов.

При переходе от системы транзакций и ресурсов к сетевой модели можно пользоваться следующими правилами. Каждый ресурс представляется позицией, причем маркировка этой позиции (простыми метками) определяет состояние ресурса. В одноканальной СМО имеется один ресурс, имеющий два состояния: "занят" и "свободен", причем начальное состояние ресурса – "свободен". Поставим этому ресурсу в соответствие позицию R сетевой модели. Маркировка $M(R) = 1$ будет свидетельствовать о незанятости ресурса, а $M(R) = 0$ – о его занятости. Маркировка $M(R) \geq 2$ является запрещенной. Переход ресурса из одного состояния в другое представляется изменением маркировки позиции R : добавление метки соответствует освобождению ресурса, а изъятие – его занятию.

В системе транзакций и ресурсов каждая транзакция представляется простой меткой или меткой с атрибутами в зависимости от того, несет она информацию или нет. Будем считать, что в нашей одноканальной СМО каждая транзакция несет информацию и, следовательно, будет представляться меткой с атрибутами.

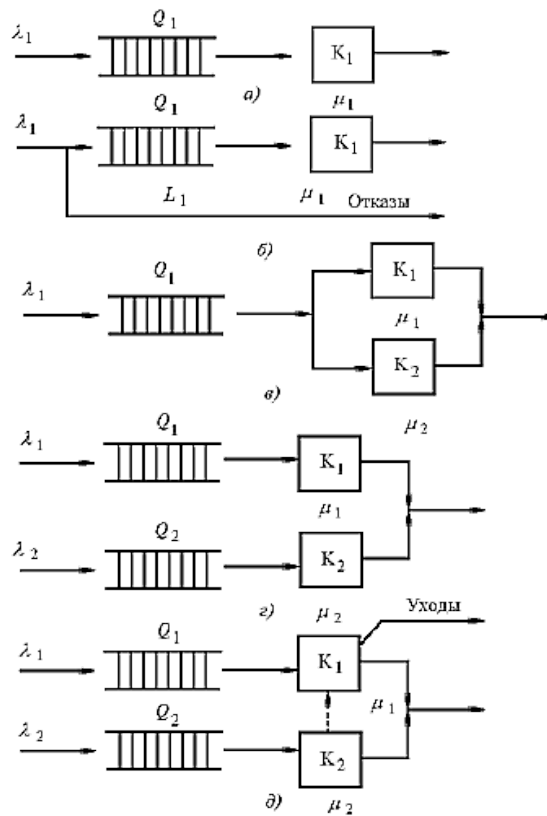


Рис. 1. Системы массового обслуживания:
a, б – одноканальные;
в, г, д – двухканальные

Транзакция в процессе ее обработки может находиться в нескольких состояниях, определяющих степень ее готовности. Для одноканальной СМО можно выделить следующие состояния транзакции:

- 0 – "готова для планирования";
- 1 – "планируется";
- 2 – "ожидание обслуживания";
- 3 – "заняла канал обслуживания";
- 4 – "обрабатывается";
- 5 – "обработана";
- 6 – "покинула канал обслуживания".

Состояния транзакции можно разделить на следующие классы:

- состояния временной активности (1, 4), связанные с процессами обработки транзакции, а также с планированием транзакции в генераторе транзакций. В этих состояниях транзакция находится во временной задержке перехода;
- состояния ожидания какого-либо ресурса (2). В этих состояниях транзакция находится в позициях, в которых нередко образуются очереди;
- состояния без ожидания (0, 3, 5, 6). В этих состояниях транзакция только "заявляет" о необходимости выполнения какого-либо события.

Каждому состоянию транзакции поставим в соответствие позицию или временную задержку перехода сетевой модели.

Переход транзакции из одного состояния в другое представляется как перемещение метки из одной позиции в другую, из позиции во временную задержку перехода, а также из временной задержки перехода в позицию.

В дальнейшем в моделируемой системе определяется множество событий, которые приводят к изменению состояния транзакций и ресурсов. В одноканальной СМО с генерацией заявок можно выделить следующие события:

1. "Приход новой заявки". При возникновении этого события транзакция переходит из состояния 1 в состояние 2. Создается копия транзакции, которая переводится в состояние 0.

2. "Планирование следующей заявки". Транзакция из состояния 0 переходит в состояние 1.

3. "Занятие канала обслуживания". Транзакция из состояния 2 переходит в состояние 3, а ресурс из состояния "свободен" переходит в состояние "занят".

4. "Начало обработки заявки". Транзакция из состояния 3 переводится в состояние 4.

5. "Конец обработки заявки". Транзакция из состояния 4 переводится в состояние 5.

6. "Освобождение канала обслуживания". Транзакция из состояния 5 переходит в состояние 6, а ресурс из состояния "занят" переходит в состояние "свободен".

7. "Уничтожение заявки". Транзакция покидает состояние 6 и уничтожается.

2. Моделирование очереди заявок

Очередью называется элемент исследуемого объекта, с помощью которого моделируются процессы ожидания начала обработки требований обслуживающими устройствами. Возникновение этих процессов происходит ввиду того, что в момент поступления требования устройство может находиться в состояниях "занято" либо "выключено". Кроме того, ожидание начала обработки может возникнуть из-за того, что приоритет поступившего требования меньше, чем приоритет обрабатываемого. По любой из вышеперечисленных причин, поступающие требования устанавливаются в очередь к устройству.

На входе устройства может формироваться произвольное количество очередей. При этом количество очередей не может превышать общего числа внесистемных и внутрисистемных источников, заданных в предложении *входящий поток*. В зависимости от количества этих источников и правил функционирования элементов, имитирующих процессы ожидания, для описания очередей используется одна из следующих конструкций:

очередь: общая / отдельная / смешанная;

С помощью первой конструкции формируется одна очередь, в которую могут устанавливаться все требования, поступающие на вход устройства. Остальные две конструкции позволяют формировать множество очередей перед устройством. Независимо от количества очередей, установленных на входе устройства, дисциплины их формирования определяются предложением *механизм обслуживания*.

Общая очередь

Общей очередью называется такая очередь, в которую могут устанавливаться требования, поступающие от всех внесистемных и внутрисистемных источников, указанных в предложении *входящий поток*. Задание общей очереди осуществляется с помощью конструкции *общая*. Формирование очереди производится независимо от способа передачи требований на вход устройства. Конкретное место, занимаемое в очереди каждым требованием, определяется в зависимости от его приоритета и механизма обслуживания. При этом основополагающим правилом является механизм обслуживания, устанавливающий дисциплину формирования очереди.

Пример модели обслуживающего устройства ($n,1$) с общей очередью представлен на рис. 1.

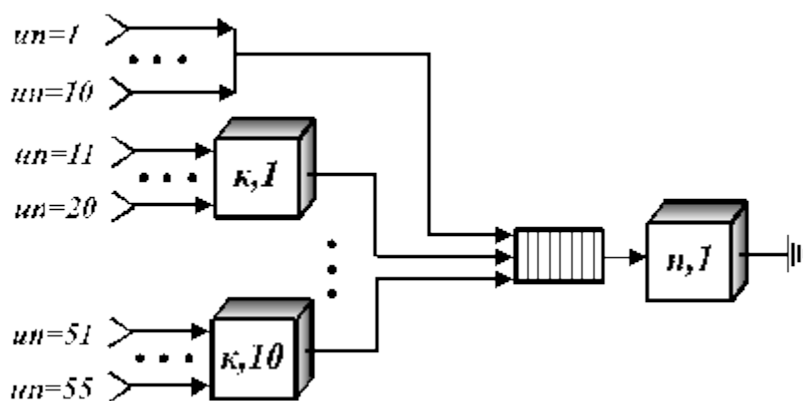


Рис. 1. Схема модели устройства с общей очередью

На вход устройства поступают требования от 10 внесистемных источников с индексами от 1 до 10 включительно и от 45 внутрисистемных источников. Интенсивность первых пяти источников требований равна 0.1, а следующих пяти источников - 0.2. Как показано на рисунке, на вход устройства $(n, 1)$ с устройства $(κ, 1)$ поступают требования от внутрисистемных источников с индексами от 11 до 20 включительно, а с устройства $(κ, 10)$ с индексами от 51 до 55 включительно. На устройствах $(κ, 2) - (κ, 9)$ обрабатываются требования, генерируемые внесистемными источниками с индексами от 21 до 50 включительно. Описание обслуживающего устройства с общей очередью можно представить следующим образом:

$у(n, 1); вп: (un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11-20 \text{ у}(κ, 1)), \dots (un=51-55 \text{ у}(κ, 10));$
од: общая; мо: впп;

С применением средств интегрированного описания, позволяющих использовать фиктивные связи, модель устройства $(n, 1)$ можно представить в виде:

$у(n, 1); вп: (un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11-55 \text{ у}(κ, 1)2-10); од: общая;$
мо: впп;

Во втором случае при наличии одного внутрисистемного входа во входящем потоке конструкция *од:общая*; также позволяет сформировать общую очередь на входе $(n, 1)$.

Отдельные очереди

Отдельными называются такие очереди, которые формируются автоматически для каждого внесистемного и внутрисистемного источника, заданного во входящем потоке.

При наличии нескольких очередей на входе устройства возникает вопрос об их приоритетах, который может решаться двумя способами: по правилу умолчания либо путем явного описания очередей в порядке убывания их приоритетов. Первый способ основывается на механизме обслуживания и порядке следования описаний внесистемных и внутрисистемных источников в предложении *входящий поток*. При этом используется конструкция *очередь: отдельная*. Установка приоритетов очередям осуществляется следующим образом. Вначале формируются очереди для всех внесистемных источников. Приоритеты этих очередей определяются порядком следования описаний внесистемных источников во входящем потоке. Затем строятся очереди для внутрисистемных источников. Приоритеты этих очередей всегда ниже приоритетов очередей, формируемых внесистемными источниками. Установка приоритетов таким очередям также осуществляется в порядке очередности появления соответствующих описаний внутрисистемных источников во входящем потоке. В процессе моделирования приоритеты очередей, установленные по правилу умолчания, могут изменяться. Подробно процедуры модификации этих приоритетов освещены при описании механизма обслуживания. Задание приоритетов очередям с использованием правила умолчания рассмотрим на примере модели, представленной на рис. 2.

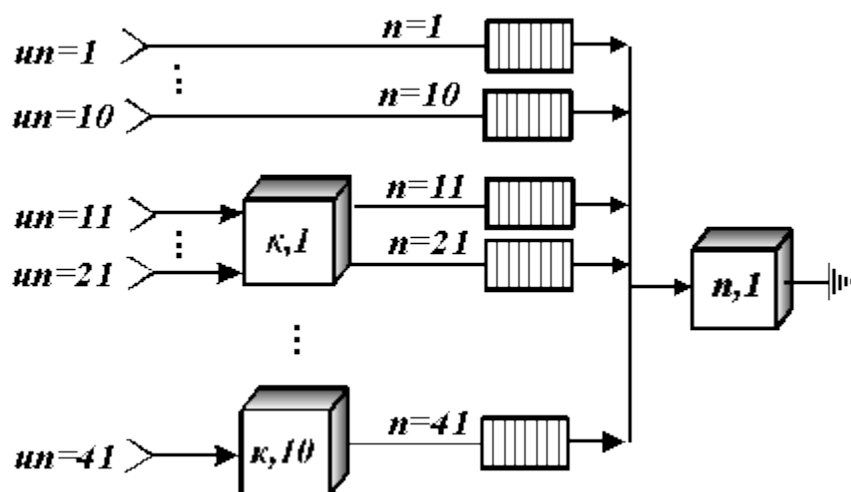


Рис. 2. Схема модели устройства с отдельными очередями

Как показано на схеме модели, на входе устройства (n, I) формируются отдельные очереди, образуемые десятью внесистемными потоками с индексами от 1 до 10 включительно и тридцатью внутрисистемными потоками с индексами от 11 до 41 включительно, поступающими с устройств $(\kappa, I) - (\kappa, 10)$. Описание этого фрагмента модели, иллюстрирующего формирование отдельных очередей, можно представить следующим образом:

$y(n, I); vn:(un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11 \text{ у}(\kappa, 1)), (un=12 \text{ у}(\kappa, 1)), \dots (un=21 \text{ у}(\kappa, 1)), \dots (un=41 \text{ у}(\kappa, 10));$

од: отдельная;

Согласно приведенному описанию на входе устройства (n, I) формируются отдельные очереди. Каждый внутрисистемный источник описывается независимо, что позволяет сформировать сорок одну очередь.

Наиболее простой способ модификации умалчиваемых значений приоритетов отдельных очередей состоит в изменении последовательности описания источников требований во входящем потоке. Предположим, что в рассматриваемом примере очередь на входе (n, I) , формируемая 41 источником требований $(un=41 \text{ у}(\kappa, 1))$, должна быть наиболее приоритетной среди очередей, образующихся из требований, поступающих от внутрисистемных источников. Для реализации такой модели описание входящего потока должно быть выполнено следующим образом:

$y(n, I); vn:(un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=41 \text{ у}(\kappa, 10)),$

$(un=11 \text{ у}(\kappa, 1)), (un=12 \text{ у}(\kappa, 1)), \dots, (un=40 \text{ у}(\kappa, 9));$

од: отдельная;

Приоритеты очередей, формируемых внесистемными источниками, также определяются согласно очередности появления соответствующих описаний. Поэтому для того, чтобы очередь, формируемая десятым внесистемным источником была наиболее приоритетной среди остальных очередей, входящий поток необходимо описать следующим образом:

$y(n, I); vn:(un=10 \text{ э}(0.2)), (un=1-5 \text{ э}(0.1)), (un=6-9 \text{ э}(0.2)), (un=41 \text{ у}(\kappa, 10)), \dots;$

При использовании правила умолчания для задания приоритетов очередям применение средств интегрированного описания внесистемных и внутрисистемных источников имеет одну особенность. Она состоит в том, что независимо от степени интеграции описания внесистемного входа для каждого из перечисленных в нем источников строятся отдельные очереди, а для внутрисистемного входа всегда создается одна очередь.

В тех случаях, когда невозможно или нецелесообразно использовать неявный способ задания приоритетов очередям, можно применить второй способ установки приоритетов. Он предполагает задание приоритетов очередям путем последовательного описания соответствующих входов в предложении *очередь*:

отдельная

((<вход 1>) [<атрибуты очереди>])
[,(<вход 2>) [<атрибуты очереди>]]
[...,((<вход N>)[<атрибуты очереди>])]]

Представленная синтаксическая конструкция позволяет первой очереди, задаваемой с помощью конструкции <вход 1>, установить наивысший приоритет, а последней очереди, <вход N> - наименьший. Например, конструкция **од: отдельная** ((*un*=4)), ((*un*=11-15 *y*(*κ*,1))), ((*un*=8-10)); устанавливает убывающие приоритеты очередям, формируемым различными типами источников в следующем порядке:

- 1 очередь - ((*un*=4));
- 2 очередь - ((*un*=11-15 *y*(*κ*,1)));
- 3 очередь - ((*un*=8));
- 4 очередь - ((*un*=9));
- 5 очередь - ((*un*=10));

Из приведенного примера видно, что для каждого внесистемного источника всегда формируется отдельная очередь, хотя в тексте модели задание третьей, четвертой и пятой очередями осуществляется с помощью интегрированного входа ((*un*=8-10)).

При построении отдельных очередей не обязательно описывать все источники требований, заданные во входящем потоке. В этом случае алгоритм построения очередей и установки им необходимых приоритетов будет основываться на использовании двух указанных выше способов. Наиболее приоритетными будут очереди, формируемые на основе их явного описания. Для неописанных источников очереди создаются по правилу умолчания. Предположим, что описание входящего потока и очереди для устройства (*n*,1) представлено следующим образом:

вп: ((*un*=11-21 *y*(*κ*,1)), ((*un*=22-24 *y*(*κ*,2)), ((*un*=25-26 *y*(*κ*,3)),
((*un*=27-28 *y*(*κ*,4)), ((*un*=29-30 *y*(*κ*,5)), ((*un*=31-32 *y*(*κ*,6)),
((*un*=33-34 *y*(*κ*,7)), ((*un*=35-36 *y*(*κ*,8)), ((*un*=37-40 *y*(*κ*,9)),
((*un*=41 *y*(*κ*,10)),((*un*=1-10 э(0.01));

од: **отдельная** ((*un*=25-26 *y*(*κ*,3))), ((*un*=1)), ((*un*=11-21 *y*(*κ*,1)));

Согласно этому описанию наивысшим приоритетом будет обладать очередь, формируемая из требований, поступающих с устройства (*κ*,3). Второй по порядку приоритет будет иметь очередь, образуемая из требований потока с индексом 1. Следующим по порядку приоритетом будет обладать очередь, формируемая из требований потоков с индексами от 11 до 21 включительно. Приоритеты остальным очередям задаются по правилу умолчания.

Смешанные очереди

Конструкция **смешанная** предназначена для формирования очередей, в которые могут устанавливаться требования, поступающие от различных типов источников. С помощью этой конструкции пользователю предоставляется возможность задавать различные правила объединения потоков для формирования очереди. Вслед за конструкцией **смешанная** необходимо указать, из каких источников должна формироваться первая очередь, затем вторая и т.д. Задание правила формирования очереди производится путем перечисления входов, образующих эту очередь. Далее может следовать описание атрибутов очереди, определяющих некоторые особенности ее функционирования. Синтаксис смешанной очереди представляется следующим образом:

смешанная

((<вход>)[,<вход>][...,<вход>] [<атрибуты очереди>]]]
[,(<вход>)[,<вход>][...,<вход>] [<атрибуты очереди>]]]
[...,((<вход>)[,<вход>][...,<вход>])] [<атрибуты очереди>]]]

Каждая из очередей, формируемых с использованием конструкции **смешанная**, может образовываться из внесистемных и внутрисистемных источников. Ниже приведен пример модели, в которой на входе устройства (*n*,1) имеются две очереди (рис. 3).

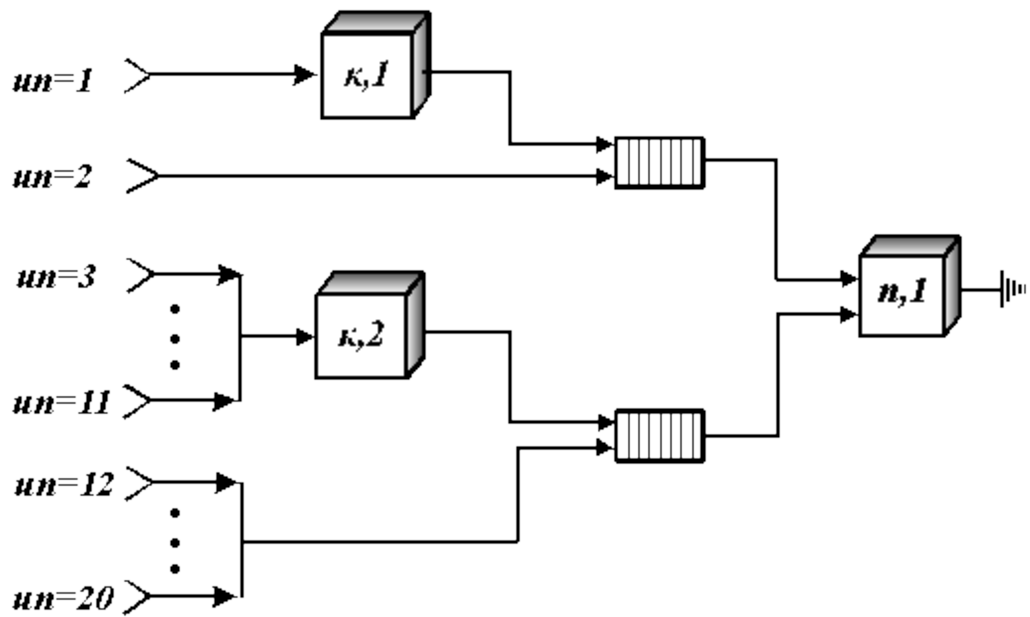


Рис. 3. Схема модели устройства со смешанными очередями

Первая из них формируется внутрисистемным потоком с индексом 1, поступающим с устройства ($\kappa,1$), и внесистемным потоком с индексом 2. Вторая очередь также образуется из различных типов источников. При этом внутрисистемные потоки с индексами 3 - 11 поступают с устройства ($\kappa,2$). Кроме того, в очередь могут устанавливаться требования, поступающие от девяти внесистемных источников с номерами 12 - 20. Описание фрагмента модели, иллюстрирующего задание смешанных типов очередей, представляется следующим образом:

$y(n,1)$; $вп$: ($un=1 y(\kappa,1)$), ($un=2 \varepsilon(0.1)$), ($un=3-11 y(\kappa,2)$), ($un=12-20 \varepsilon(0.2)$);
од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-20$)));

Если в описании смешанной очереди перечислены не все входы, то при построении очередей по умолчанию для каждого из них создается отдельная очередь. Предположим, что в рассмотренной выше модели для девятнадцатого и двадцатого потоков необходимо построить отдельные очереди. Для этого достаточно при описании правила формирования второй очереди опустить в перечислении девятнадцатый и двадцатый потоки:

$y(n,1)$; $вп$: ($un=1 y(\kappa,1)$), ($un=2 \varepsilon(0.1)$), ($un=3-11 y(\kappa,2)$), ($un=12-20 \varepsilon(0.2)$);
од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-18$)));

Этой же цели можно достигнуть путем прямого задания правила формирования третьей и четвертой очередей для 19-го и 20-го потоков:

$y(n,1)$; $вп$: ($un=1 y(\kappa,1)$), ($un=2 \varepsilon(0.1)$), ($un=3-11 y(\kappa,2)$), ($un=12-20 \varepsilon(0.2)$);
од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-18$)), (($un=19$)), (($un=20$)));

Если необходимо, чтобы 19 и 20-ый потоки образовывали одну очередь, то описание очереди должно быть представлено следующим образом:

од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-18$)), (($un=19$), ($un=20$)));

Как и в случае с отдельным типом очереди, при наличии более одной очереди на входе устройства, возникает вопрос об их приоритетах. Приоритет задается согласно очередности описания правила формирования очереди. Первая очередь, встретившаяся в описании, имеет наивысший приоритет. Приоритет остальных очередей последовательно убывает. Если при описании правил формирования смешанной очереди указаны не все входы, перечисленные во входящем потоке, то для них по умолчанию также создаются отдельные очереди. При этом сначала строятся очереди для внесистемных, а затем - для внутрисистемных потоков.

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

6.1.1. Основная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л1.1	ДГТУ, Каф. ИСвС"; сост.: Г.Б. Анисимова, Е.Е. Позднышева	Методы и средства проектирования информационных систем и технологий. Структурный подход: метод. указания	Ростов н/Д.: ИЦ ДГТУ, 2018	https://ntb.donstu.ru/content/methody-i-sredstva-proektirovaniya-informatsionnyh-sistem-i-tehnologiy-strukturnyy-podhod-metodicheskie-ukazaniya
Л1.2	Водяхо А. И., Выговский Л. С., Дубенецкий В. А., Цехановский В. В.	Архитектурные решения информационных систем	, 2017	https://e.lanbook.com/book/96850
Л1.3	Долженко А. И.	Технологии командной разработки программного обеспечения информационных систем	Москва: Интернет - Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/39569.html
6.1.2. Дополнительная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л2.1	Флегонтов А. В., Матюшичев И. Ю.	Моделирование информационных систем. Unified Modeling Language: учебное пособие	, 2019	https://e.lanbook.com/book/112065
Л2.2	Васильев Р. Б., Калянов Г. Н., Лёвочкина Г. А.	Управление развитием информационных систем	Москва: Интернет - Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/62828.html
Л2.3	Трофимов Г. Д.	Разрешение конфликтов в трудовых коллективах	Москва: Лаборатория книги, 2010	http://biblioclub.ru/index.php?page=book&id=88658
Л2.4	Овчинников А. Л.	Исследование интеллектуальной активности коллектива	Москва: Лаборатория книги, 2011	http://biblioclub.ru/index.php?page=book&id=140319



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Организация, управление, планирование и
прогнозирование научных исследований» для студентов
направления подготовки
09.04.02 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Организация, управление, планирование и прогнозирование научных исследований» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии, направленность (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1

Лабораторная работа 2

Лабораторная работа 3

Лабораторная работа 4 .

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины – освоение методов разработки математических моделей информационных процессов и методологии и технологии математического моделирования при исследовании, проектировании, эксплуатации информационных систем; формирование общекультурных и профессиональных компетенций магистра в соответствии с требованиями ФГОС по направлению Информационные системы и технологии; подготовка магистра к деятельности, требующей применения научно-практических знаний и умений в области анализа информационных процессов; развитие логического, алгоритмического мышления студентов, умения самостоятельно расширять свои знания в области математического представления информационных процессов.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

В результате освоения данной дисциплины формируется следующая компетенция у обучающегося:

ОПК-3.2: Анализирует профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров

УК-4.2: Применяет на практике коммуникативные технологии, методы и способы делового общения для академического и профессионального взаимодействия

УК-3.2: Разрабатывает план групповых и организационных коммуникаций при подготовке и выполнении проекта

УК-1.2: Соотносит разнородные явления и систематизирует их в рамках избранных видов профессиональной деятельности

Изучив данный курс, студент должен:

Знать:

методику априорного ранжирования факторов, метод наименьших квадратов, методы математического моделирования, методы оптимизации параметров иметь представление о математических моделях технических систем и о применении методов математического моделирования для исследования технических объектов.

Уметь:

самостоятельно работать с учебной, справочной и учебно-методической литературой; использовать методику априорного ранжирования факторов, применять метод наименьших квадратов, методы оптимизации параметров и методы математического моделирования; применять численные методы для решения задач с использованием прикладных математических пакетов.

Владеть:

учебной и учебно-методической литературой; навыками проведения экспериментальных исследований; навыками обработки и анализа результатов эксперимента; методом математического моделирования.

Реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения

практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

«Умение ставить разумные вопросы есть уже важный и необходимый признак ума и проницательности» (И. Кант)

Для успеха научного исследования его необходимо правильно организовать, спланировать и выполнять в определенной последовательности.

Научно-исследовательские организации и образовательные учреждения разрабатывают планы исследовательской работы на год на основе комплексных программ, долгосрочных научных и научно-технических программ, договоров, заявок, грантов на исследования, представленных заказчиками. Научная работа кафедр учебных заведений организуется и проводится в соответствии с планами работы на учебный год. Профессора, преподаватели и аспиранты выполняют научно-исследовательские работы по индивидуальному плану. Планируется и научно-исследовательская работа студентов. Планы работы учебных заведений и кафедр могут содержать соответствующий раздел о НИР студентов. По планам работают студенческие научные кружки и проблемные группы.

Для того, чтобы упорядочить основные этапы научно-исследовательской работы в соответствии с программой исследования, календарными сроками и материальными затратами, составляется рабочий план (план-график) выполнения работ. Необходимо так выстроить логическую очередность выполнения работ, чтобы она в установленные сроки привела к достижению поставленной цели и решению научной задачи. В работе необходимо выделить главное, на чем следует сосредоточить внимание в данный момент, но вместе с тем нельзя упускать из поля зрения детали.

План и последовательность действий зависят от вида, объекта и целей научного исследования, но, в общем, схема научного исследования выглядит следующим образом:

1. формулирование цели и задачи;
2. выбор объекта и материалов, используемых для эксперимента;
3. выбор методики проведения эксперимента;
4. установление точности результатов измерений (выходных параметров),
5. фиксация и обработка результатов эксперимента
6. обсуждение и выводы

Сначала формулируется тема научного исследования, обосновываются причины её разработки. После предварительного ознакомления с литературой и материалами ранее проведенных исследований выясняется, в какой мере вопросы темы изучены и каковы полученные результаты.

Тема научной работы должна отвечать некоторым требованиям:

- 1) актуальность;
- 2) конкретность;
- 3) проблемность;
- 4) определенность понятий;
- 5) краткость.

На первом месте здесь критерий актуальности научной темы, который определяется новизной, связью темы с жизнью общества и так называемой модностью. Если тема модная,

оперативно отражает реальность с ее проблемами, то можно выиграть, если раньше конкурентов написать, опубликовать работу, в которой предложить решение этой проблемы. Но если промедлить, то появляется огромное количество конкурентов, и будет сложно показать преимущества и новизну своей работы, и ее оригинальность.

Второе условие — конкретность. Типичные недостатки: неоправданная широта темы, абстрактность, когда формулируются темы глобального характера.

Третье условие — наличие проблемы.

Четвертое условие — определенность основных понятий, недопустимость употребления двусмысленных и нечетких выражений, например, «рост развития», «уменьшение смысла», «философия проблемы», «магия привлекательности».

Пятое условие — краткость. В теме может и должен быть отражен объект и предмет исследования. В соответствии с обозначенной темой формулируется цель, которая должна быть ясно и четко выражена.

Требования к формулированию темы (названия) научной работы:

- 1) в названии темы должна быть сформулирована проблема;
- 2) тема должна быть достаточно конкретной;
- 3) в названии темы должны присутствовать важнейшие категории, выражающие качественные характеристики проблемы.

Цель и задачи

Цель исследования – это то, к чему стремится исследователь в своей работе, его общая направленность на конечный результат. Цель работы обычно созвучна названию темы научного исследования. Целью работы может быть описание нового явления, изучение его характеристик, выявления закономерностей и т. д.

В предметном поле исследователь избирает проблему, неразрешенную трудность, которая мешает нормальному развитию общества и успешному удовлетворению его потребностей, улучшению экологической ситуации и т. д. Решение этой проблемы и рассматривается им как цель его работы. Цель выступает как конечная причина, которая определяет характер, ход и задачи исследования. Отсюда вытекает логичность структуры работы, которая должна быть подчинена конкретной цели.

Далее, в соответствии с поставленной целью определяется круг задач. Задачи должны как бы логически расчленять цель на первичные составные элементы. Основопологающим требованием при этом является выделение логического основания, по которому будет произведена классификация задач. Чаще всего в качестве основания классификации задач выдвигают этапы исследовательской работы. Каждому этапу обычно посвящается отдельная задача. В перечне решаемых задач необходимо выделять наиболее крупные без их дробления на более мелкие задачи. Задачи исследования – это то, что требует решения в процессе исследования, вопросы, на которые должен быть получен ответ.

Задачи исследования лучше формулировать в форме перечисления, применяя глагольные сказуемые: "проанализировать ...", "осуществить ..."; "раскрыть особенности организации...", "выделить условия развития..."; "выявить совокупность положений, составляющих теоретико-методологические основания...", "исследовать ...", "обосновать эффективность...", "определить принципы ..."; "разработать ..."; "апробировать...".

Объект и предмет исследования

Объект исследования в определении теории познания – это то, что изучает эта наука, что противостоит познающему субъекту в его познавательной деятельности. То есть, это та

окружающая действительность, с которой исследователь имеет дело.

Предмет исследования – это та сторона, аспект, точка зрения, «проекция», с которой исследователь познает целостный объект, выделяя при этом главные, наиболее существенные (с точки зрения исследователя) признаки объекта. Другими словами – это стороны, особенности объекта, которые подлежат изучению. Например, если тема научной работы посвящена биотехнологии растений, то объектом исследования является процесс жизнедеятельности растительной клетки, а предметом – способы и методы модификации клеточных компонентов с целью получения желаемого эффекта.

Следует отметить, что один и тот же объект может быть предметом разных исследований или даже целых научных направлений. Например, кристалл каменной соли. Кристаллография изучает его с точки зрения оптических характеристик, механических, может быть, пьезоэлектрических. Химия исследует его состав, реакции, время растворения. Минералогия рассматривает его мутность, включения, выясняет происхождение и условия формирования.

ПРОВЕДЕНИЕ НАУЧНОГО ИССЛЕДОВАНИЯ

Кроме грамотного планирования, очень важно так провести исследование (эксперимент) и оформить его результаты, чтобы оно было адекватно воспринято другими исследователями и в случае необходимости, его можно было повторить и подтвердить приоритет данного исследователя или лаборатории. Поэтому, после теоретического планирования наступает преаналитический этап, где происходит отбор методик для эксперимента, определяется необходимое число измерений и количество экспериментальных групп, готовится необходимое оборудование, химические реагенты и посуда. Это очень важный этап, определяющий дальнейший успех или неудачу всего исследования, поэтому требует тщательной проработки. Для успешного осуществления этого этапа в биохимических, гистологических, микробиологических и других экспериментах, исследователю необходимо знание методов отбора и хранения анализируемых образцов, маркировки химических реактивов, правил работы с основным лабораторным оборудованием и измерительными приборами, видов, степени точности и предназначения лабораторной посуды, а также правил приготовления и расчета основных рабочих реактивов. Однако кроме технических моментов очень важной составляющей любого исследования является интерпретация полученных данных. Вся последовательность событий при медико-биологическом эксперименте может быть представлена в виде континуума, состоящего из двух уровней. Первый уровень начинается с измерения характеристик объекта и заканчивается получением результата путем сравнения значений со специальными шкалами. Второй уровень – синтетический, он базируется на сравнении результатов с литературными данными, научными гипотезами и т. п. Результатом этого этапа являются научно обоснованные выводы и практические рекомендации. При их формулировании исследователи очень часто используют приемы по упрощению данных полученных при измерениях. Результаты измерения чаще всего представляют собой количественные непрерывные данные, в то время как выводы содержат информацию в виде качественных, дискретных данных. Поэтому представляется важным рассмотреть пошагово весь ниже приведенный континуум.

В клинической лабораторной диагностике общепринятым считается деление исследования на 3 этапа, а именно преаналитический, аналитический и постаналитический. Это в первую очередь обусловлено тем, что первый и третий этапы трудно стандартизировать и поэтому на них совершается наибольшее количество ошибок, ведущих к ложным клиническим решениям. В клинических исследованиях по сравнению с медико-биологическим экспериментом на преаналитическом этапе могут выполняться многочисленные операции и естественно, что любые отклонения в технологии скажутся на конечном результате. Аналитический этап включает непосредственно измерение, и интерпретацию величины этого измерения в соответствии со специализированной шкалой при этом исследователь получает результат сопоставимый с литературными данными. Постаналитический этап включает интерпретацию этого результата. Обычно на этом этапе проводится упрощение данных и приведение количественных данных к качественным – дискретным, что позволяет принять клиническое решение или дать рекомендацию.

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

6.1.1. Основная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л1.1	Леонова, О. В.	Основы научных исследований: учебное пособие	Москва: Московская государственная академия водного транспорта, 2015	http://www.iprbookshop.ru/46493.html
Л1.2	Сафин, Р. Г., Иванов, А. И., Тимербаев, Н. Ф.	Основы научных исследований. Организация и планирование эксперимента: учебное пособие	Казань: Казанский национальный исследовательский технологический университет, 2013	http://www.iprbookshop.ru/62219.html
6.1.2. Дополнительная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л2.1	Бойко, А. Ф., Воронкова, М. Н.	Теория планирования многофакторных экспериментов: учебное пособие	Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2013	http://www.iprbookshop.ru/28403.html
Л2.2	Сагдеев, Д. И.	Основы научных исследований, организация и планирование эксперимента: учебное пособие	Казань: Казанский национальный исследовательский технологический университет, 2016	http://www.iprbookshop.ru/79455.html
6.1.3. Методические разработки				
	Авторы,	Заглавие	Издательство, год	Адрес
Л3.1	Бекряев, В. И.	Практикум по основам теории эксперимента	Санкт-Петербург: Российский государственный гидрометеорологический университет, 2003	http://www.iprbookshop.ru/12520.html
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"				
Э1	Порсев, Е. Г. Организация и планирование экспериментов [Электронный ресурс] : учебное пособие / Е. Г. Порсев. — Электрон. текстовые данные. — Новосибирск : Новосибирский государственный технический университет, 2010. — 155 с. — 978-5-7782-1461-3. — Режим доступа: http://www.iprbookshop.ru/45415.html			
Э2	Попов, А. А. Оптимальное планирование эксперимента в задачах структурной и параметрической идентификации моделей многофакторных систем [Электронный ресурс] : монография / А. А. Попов. — Электрон. текстовые данные. — Новосибирск : Новосибирский государственный технический университет, 2013. — 296 с. — 978-5-7782-2329-5. — Режим доступа:			
Э3	Полякова, Н. С. Математическое моделирование и планирование эксперимента [Электронный ресурс] : методические указания к выполнению домашнего задания / Н. С. Полякова, Г. С. Дерябина, Х. Р. Федорчук. — Электрон. текстовые данные. — М. : Московский государственный технический университет имени Н.Э. Баумана, 2010. — 36 с. — 2227-8397. — Режим доступа:			



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Математические модели информационных
процессов» для студентов направления подготовки
09.04.02 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Математические модели информационных процессов» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии, направленность (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1 Исследование модели непрерывно-детерминированной системы

Лабораторная работа 2 Исследование стохастической имитационной модели системы

Лабораторная работа 3 Исследование элементов модели системы массового обслуживания

Лабораторная работа 4 Исследование модели системы массового обслуживания

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины – освоение методов разработки математических моделей информационных процессов и методологии и технологии математического моделирования при исследовании, проектировании, эксплуатации информационных систем; формирование общекультурных и профессиональных компетенций магистра в соответствии с требованиями ФГОС по направлению Информационные системы и технологии; подготовка магистра к деятельности, требующей применение научно-практических знаний и умений в области анализа информационных процессов; развитие логического, алгоритмического мышления студентов, умения самостоятельно расширять свои знания в области математического представления информационных процессов.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

В результате освоения данной дисциплины формируется следующая компетенция у обучающегося:

ОПК-6.2: Оценивает процессы получения, передачи, хранения и представления информации на основе положений системной инженерии.

Изучив данный курс, студент должен:

Знать:

Инструментальные средства математического моделирования информационных процессов и условия их применимости.

Теоретические основы функционирования информационных систем и процессов и математический аппарат для их описания.

Уметь:

Создавать и исследовать математические модели информационных процессов с использованием стандартных пакетов автоматизированного проектирования.

Владеть:

Созданием математических моделей информационного процесса на языке высокого уровня и с использованием интегрированных пакетов прикладных программ.

Оценки адекватности математической модели.

Реализация компетентного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа 1 Исследование модели непрерывно-детерминированной системы

Целью занятия является освоение методов моделирования непрерывно-детерминированных динамических систем. В процессе выполнения задания студенты должны:

разработать модели заданной системы, реализованные в пакете визуального моделирования Simulink на основе полученного описания;

получить результаты исследования данных моделей и выполнить их объяснение в терминах прикладной области.

В качестве примера учебного задания рассматривается задача об управлении протеканием воды через систему, состоящую из двух цилиндрических баков, расположенных на разной высоте, или, более кратко, задача о двух баках. Выбор этой задачи в качестве примера обусловлен следующими причинами:

понятным физическим принципом;

нелинейным поведением компонент системы;

наличием аварийных ситуаций в поведении системы, которые необходимо

обрабатывать;

Модель системы, реализованная в подсистеме Simulink пакета Matlab

Построение модели в Simulink сводится к перемещению с помощью мыши необходимых блоков из библиотек Simulink в окно создаваемой модели и соединению этих блоков между собой с помощью функциональных связей. Библиотеки Simulink содержат большое количество разнообразных функциональных блоков, которые отображаются на экране в виде пиктограмм.

Используя описание задачи о двух баках можно выделить в данном примере подсистему System (рис.3), представляющую собой составной блок, который содержит в себе функциональную схему, содержащую в себе диаграмму Stateflow (представленную блоком Controller, являющимся экземпляром стандартного блока Chart и описывающую поведение контроллера) и составной блок Tank_System_Block, соединенные соответствующими функциональными связями (рис.4). Так же в блоке System присутствуют часы Clock, подающие системное время из Simulink в диаграмму Stateflow. Это объясняется тем, что при построении моделей, где используются блоки Stateflow (содержащие в себе переходы, инициируемые истечением неких временных интервалов) необходимо синхронизировать внутреннее системное время в Stateflow и в Simulink. Для этого необходимо на отдельный вход в блоке Stateflow подавать системное время из Simulink и именно это время использовать при составлении условий переходов (таких как истечение отрезков времени Time1 и Time2 в задаче о двух баках). Как видно из рис.4, к связи, соединяющей блок Clock и Controller, подсоединены два экземпляра блока Hit Cross. Еще два экземпляра блока Hit Cross подсоединены к связи блока Controller и выхода h2 блока Tank_System_Block. Использование Hit Cross блоков необходимо для правильного выполнения переходов в диаграмме Stateflow, включенной в модель

Simulink, в которой происходит непрерывное интегрирование. На вход Hit Cross блока подается некая величина (в модели двух баков для двух Hit Cross блоков это системное время, для еще двух - величина h2). Сам Hit Cross блок содержит в себе некую величину, при совпадении которой с величиной, подаваемой на вход Hit Cross блока, система уменьшает шаг интегрирования. В связи со спецификой системы Stateflow это влияет на правильное выполнение переходов и не позволяет системе «проскочить момент», когда должен выполняться тот или иной переход. У первого Hit Cross блока внутренней величиной является момент времени Time1, у второго - Time 1+Time2, у третьего - значение L_plus, у четвертого - L_minus.

Блок System не имеет входов и имеет два выхода - h1 и h2, соединенные со стандартным блоком Mux (объединяющим их в вектор (h1,h2)), соединенный со стандартным блоком вывода Scope (на который он подает вектор выходных величин (h1,h2)).

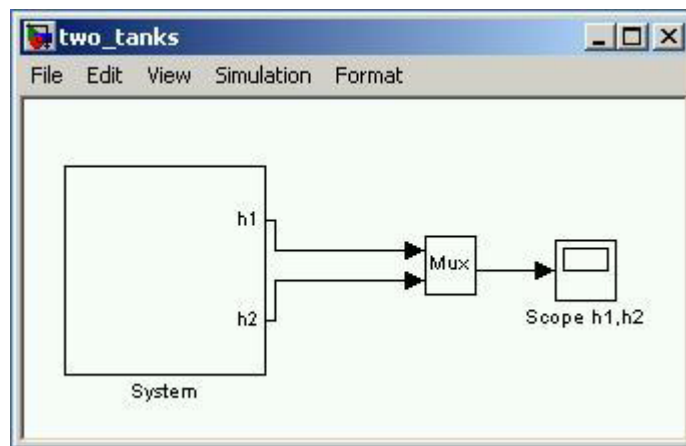


Рисунок 3

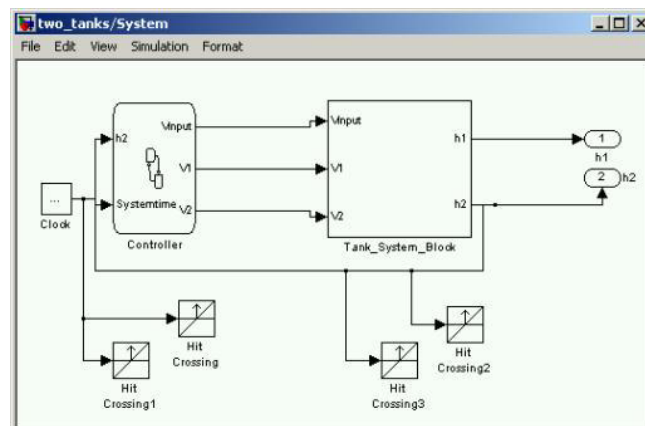


Рисунок 4

В свою очередь, составной блок Tank_System_Block содержит в себе функциональную схему, состоящую из составного блока Tank_System (имеющего три входа, соединенные с соответствующими выходами блока Chart, на которые подаются сигналы контроллера, и два выхода - h1 и h2) и пар экземпляров стандартных блоков Const, Relational Operator и Stop Simulation (выполняющих проверку на переполнение баков), соединенных соответствующими функциональными связями (рис.5). Блок Tank_System_Block имеет три входа (на которые подаются сигналы от контроллера) и два выхода (h1 и h2).

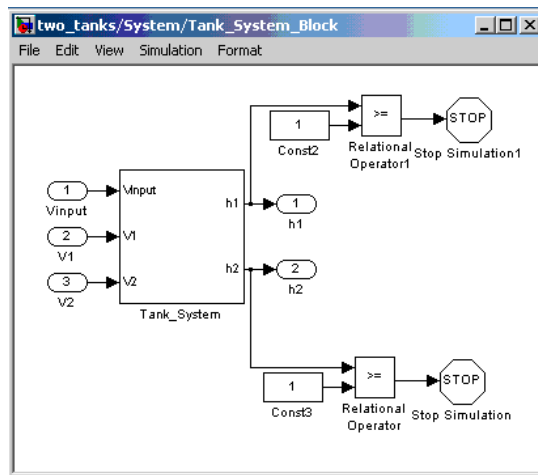


Рисунок 5

Составной блок Tank_System содержит в себе функциональную схему, состоящую из составных блоков Two_Tanks, Vin_Control, K1_Control и K2_Control, соединенных между собой функциональными связями (рис.6). Блок Tank_System имеет то же количество входов и выходов с теми же значениями, что и предыдущий блок.

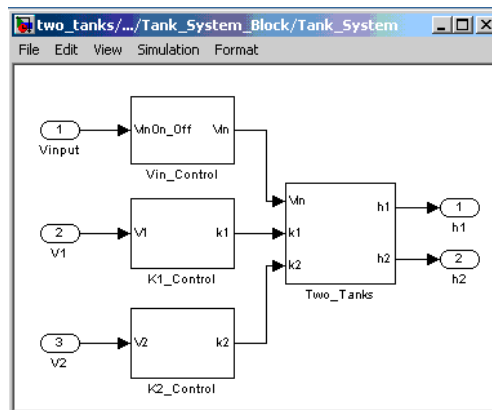


Рисунок 6

Составной блок Vin_Control содержит в себе функциональную схему, состоящую из стандартного блока Switch и двух экземпляров стандартного блока Const, содержащих в себе значения, соответствующие скорости входного потока воды в систему двух баков, когда входной кран открыт и закрыт. Переключения между этими значениями происходит в блоке Switch в зависимости от значения, поступающего на вход блока Vin_Control от контроллера. Соответствующее значение Vin подается на выход. Блок Vin_Control имеет один вход и один выход (рис.7):

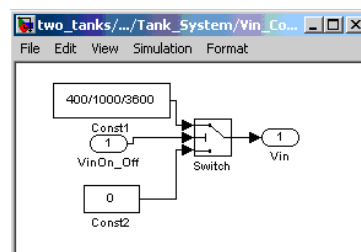


Рисунок 7

Составные блоки K1_Control и K2_Control содержат в себе идентичные функциональные схемы, различающиеся только значением коэффициентов в уравнении, представленном в стандартном блоке fcn. Функциональная схема состоит из двух экземпляров стандартного блока Switch (один из которых необходим для переключения между положениями кранов V1/V2 открыто/закрыто, а второй для отслеживания ситуации, когда параметр p становится равным 80 и соответствующего переключения между значениями функции K(p)), трех экземпляров стандартного блока Const (содержащих в себе значения, 1 и -1, соответствующие положению крана V1/V2 (открывается/закрывается) и значение 0 для функции K(p) в ситуации, когда $p \geq 80$), стандартного блока Integrator (интегрирующего поступающее от переключателя значение в пределах от 0 до 80 с начальным значением интегрируемой величины 80) и стандартного блока f(u), в котором происходит вычисление значения функции K1(p)/K2(p). Переключения в блоке Switch происходит в зависимости от значения, поступающего на вход блока K1_Control/K2_Control от контроллера. Соответствующее значение k1/k2 подается на выход. Блок K1_Control/K2_Control имеет один вход и один выход (на рис.8 представлен блок K1_Control, блок K2_Control идентичен):

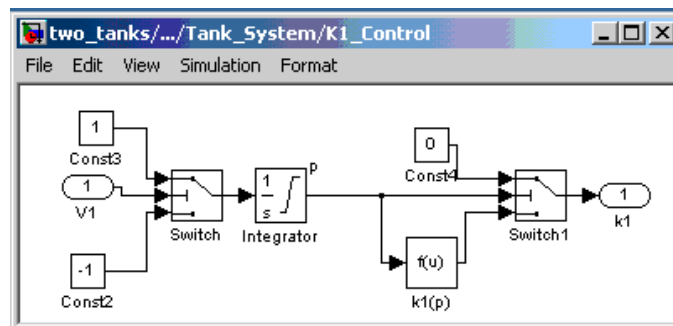


Рисунок 8

Составной блок Two_Tanks содержит в себе функциональную схему, состоящую из составных блоков Vout_calc, V12_calc1 и V12_calc2 (отвечающих за вычисление значений Vout и V12), стандартного блока Switch, двух экземпляров стандартного блока Mux (объединяющих в вектора соответствующие значения - Vin (поступающее на вход блока Two_Tanks с выхода блока Vin_Control)) и V12, а также V12 и Vout), двух экземпляров стандартного блока fcn (производящих вычисление производных h1 и h2), двух экземпляров стандартного блока Integrator, выходами которых являются значения h1 и h2, поступающие на выход. Переключения в блоке Switch происходит в зависимости от значения h2, поступающего на вход блока Switch от интегратора, и в зависимости от этого значения, подключается один из блоков, вычисляющих V12. Блок Two_Tanks имеет три входа, на которые подаются значения Vin, k1 и k2; и два выхода - h1 и h2 (рис.9):

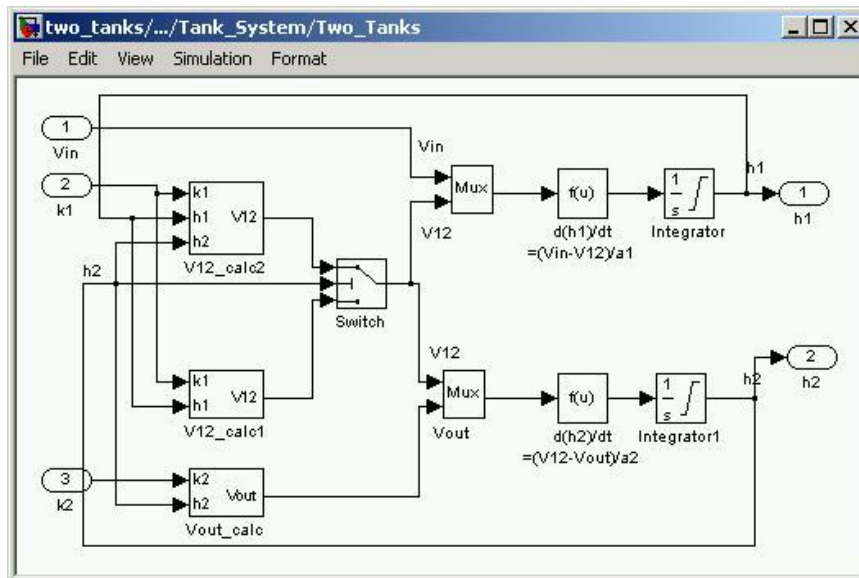


Рисунок 9

Составной блок Vout_calc содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k2 и h2, поступающие на вход блока Vout_calc, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины Vout. Блок Vout_calc имеет два входа и один выход (рис.10):

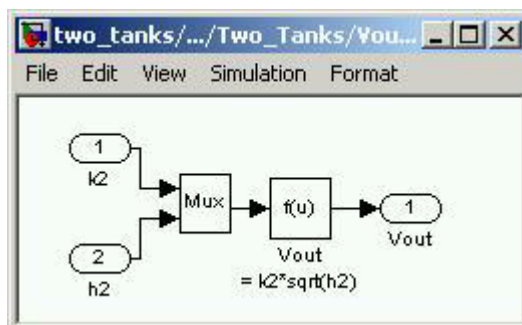


Рисунок 10

Составной блок V12_calc1 содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k1 и h1, поступающие на вход блока V12_calc1, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины V12. Блок V12_calc1 имеет два входа и один выход (рис.11):

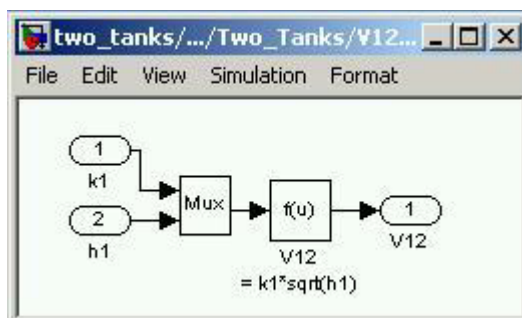


Рис. 11

Составной блок V12_calc2 содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k1, h1 и h2, поступающие на вход блока V12_calc2, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины V12. Блок V12_calc2 имеет три входа и один выход (рис.12):

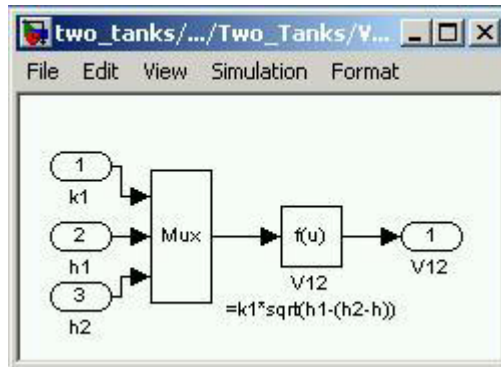


Рис. 12

Диаграмма Stateflow, представлена экземпляром стандартного блока Chart, (рис.13). В ней вместо внутреннего времени t используется подаваемое на вход блока системное время Simulink (входная переменная Systemtime).

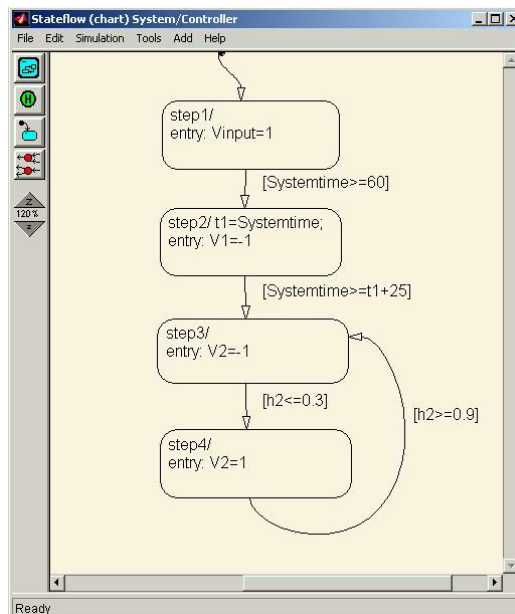


Рис. 13

Результаты эксперимента

На рис.14 представлена временная диаграмма изменений уровней воды в баках, полученная в результате эксперимента с моделью, реализованной в подсистеме Simulink пакета Matlab:

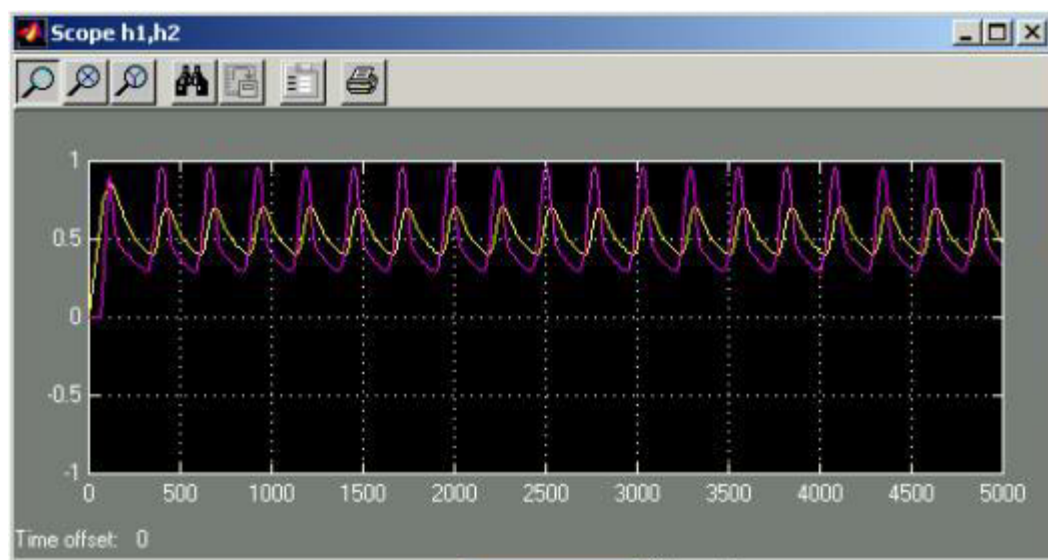


Рис. 14

Данные временные диаграммы отражают изменения уровней воды в системе двух баков при открытии/закрытии выходного крана. Возрастающие участки кривых $h_1(t)$ и $h_2(t)$ соответствуют состоянию системы, когда выходной кран закрыт. Убывающие участки кривых $h_1(t)$ и $h_2(t)$ соответствуют состоянию системы, когда выходной кран открыт.

Лабораторная работа №2 Исследование стохастической имитационной модели системы

Целью занятия является практическое освоение методов моделирования стохастических динамических систем. В процессе выполнения задания студенты должны:

разработать модель заданной системы с использованием пакета визуального моделирования Simulink на основе полученного описания;

получить результаты исследования модели и выполнить их объяснение в терминах прикладной области.

1. Описание моделируемой системы

В практике передачи и приема информации с использованием радиоканалов связи одной из важнейших задач является оценка влияния помех на вероятность безошибочной передачи информации. Рассмотрим систему цифрового радиоканала, представленную на рис. 1.

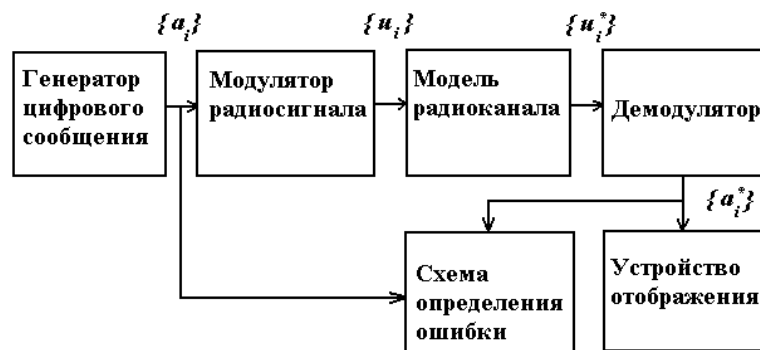


Рисунок 1

Генератор цифровых сообщений служит для имитации двоичных кодов заданной длины и может быть представлен следующим выражением

$$\begin{cases} a_i = 0, \text{ если } Rnd(i) < 0,5 \\ a_i = 1, \text{ если } Rnd(i) \geq 0,5 \end{cases}$$

где $Rnd(i)$ число из интервала $0 \dots 1$, выдаваемое встроенным программным генератором псевдослучайных чисел; $i = \overline{1, I_{зад}}$.

Несущий радиосигнал задан выражением

$$u(t) = U \sin(2\pi f_n t + \varphi_0),$$

где U амплитуда радиосигнала; f_n - несущая частота; φ_0 - начальная фаза.

Модуляция радиосигнала по закону передаваемого сообщения может быть выполнена с помощью импульсной модуляции (ИМ):

$$\begin{cases} U = 0, \text{ если } a_i = 0; \\ U = 1, B, \text{ если } a_i = 1. \end{cases}$$

Частотной модуляции (ЧМ):

$$\begin{cases} f = f_n, \text{ если } a_i = 0; \\ f = f_n + \Delta f, \text{ если } a_i = 1. \end{cases}$$

Фазокодовой манипуляции (ФКМ):

$$\begin{cases} \varphi_0 = 0, \text{ если } a_i = 0; \\ \varphi_0 = \pi, \text{ если } a_i = 1. \end{cases}$$

Модель канала характеризуется видом помех. Для аддитивных помех, приводящих к искажению амплитуды сигнала, используется выражение

$$u^*(t) = u(t) \pm Rnd(t) \cdot U_{\Pi},$$

где U_{Π} - максимальное значение амплитуды помехи.

Модель канала с фазовыми искажениями имеет вид

$$u^*(t) = u(t) + U_{\Pi} \cdot \sin(2\pi f_n t \pm Rnd(t) \cdot \varphi_0).$$

Модель канала с преобладающими частотными искажениями может быть задана выражением

$$u^*(t) = u(t) + U_{\Pi} \cdot \sin(2\pi t(f_n \pm Rnd(t) \cdot \Delta f)).$$

Демодуляция определяется видом модуляции радиосигнала и для амплитудной импульсной модуляции задается выражением

$$\begin{cases} u^*(t) > U_{порога}^1, \text{ то } a_i^* = 1; \\ u^*(t) < U_{порога}^0, \text{ то } a_i^* = 0, \text{ иначе } a_i^* = x. \end{cases}$$

Пороговые значения нулевого и единичного уровней определяются исходя из условий распространения сигнала и дальности связи. Состояние x характеризует появление ошибки. Аналогичные выражения только с пороговыми значениями частоты или фазы сигнала используются для демодуляции ЧМ и ФКМ сигналов.

Фиксация ошибки осуществляется посредством сравнения $a_i = a_i^*$ или по наличию ошибки демодуляции $a_i^* = x$.

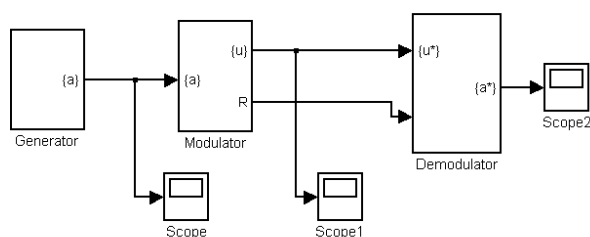
2. Задание для самостоятельной работы

В соответствии с номером по журнальному списку выбрать из табл. 1 исходные данные для моделирования и составить модель радиоканала. При составлении модели максимально использовать возможности пакета имитационного моделирования Simulink. Оформить отчет по практическому занятию в котором отразить текст описания системы и разработанной модели. Привести графики на выходе элементов модели для различных вариантов исходных данных. Оценить влияние различных помеховых воздействий на число ошибок.

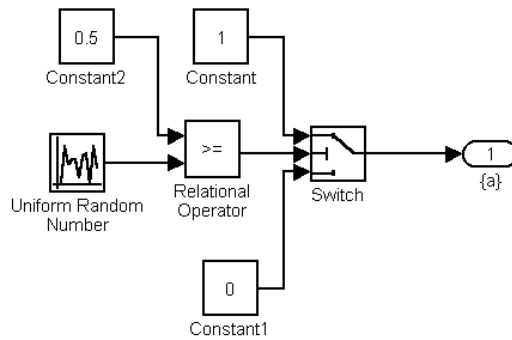
Таблица 1

№ п/п	Исходные данные для моделирования					Примечание
	Вид модуляции	Число разрядов	$U^1_{порога}$	$U^0_{порога}$	$U_{П}$	
1	ИМ	7	$0,6U$	$0,1U$	$0,01U$	
2	ФКМ	8				
3	ЧМ	9				$\Delta f = 0,1 f$
4	ИМ	10	$0,8U$	$0,2U$	$0,02U$	
5	ЧМ	7				$\Delta f = 0,2 f$
6	ФКМ	8				
7	ЧМ	9				$\Delta f = 0,01 f$
8	ИМ	10	$0,9U$	$0,1U$	$0,01U$	
9	ФКМ	7				
10	ИМ	8	$0,5U$	$0,05U$	$0,01U$	

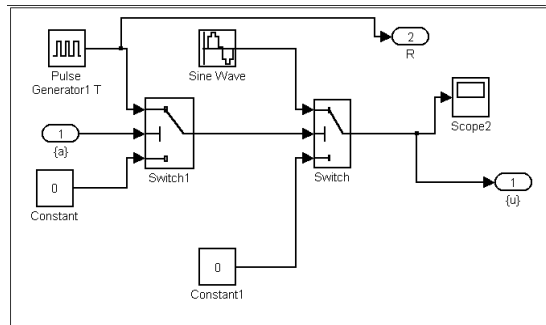
Структура модели



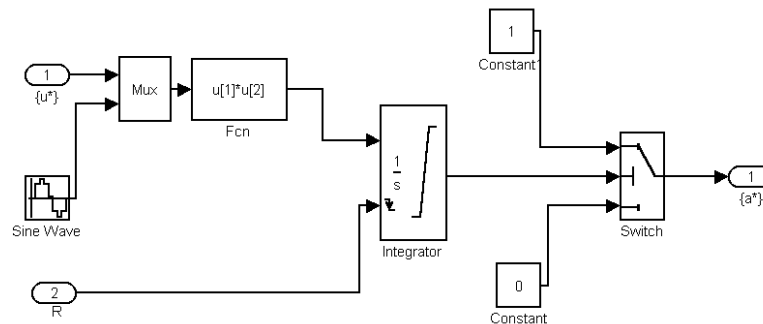
Модель генератора цифрового сигнала



Модель модулятора радиосигнала



Модель демодулятора радиосигнала



Исходные данные моделей

Время моделирования 1000.

Генератор случайных чисел:

Минимальное значение случайного числа =0; максимальное значение =1; начальное значение последовательности (любое целое число); Длительность одной реализации =100.

Minimum=0; Maximum=1; Initial seed=0; Sample time=100.

Генератор импульсов:

Период =100 секунд модели; Длительность импульса =50 (скважность равна 2); амплитуда 1 вольт; начальное время генерации =0.

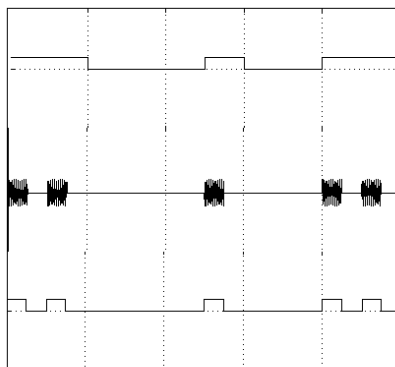
Period=100(secs); Duty cycle (% of period)=50; Amplitude=1; Start time=0.

Генератор синусоиды:

Амплитуда 1 вольт; Циклическая частота=10 радиан в секунду; Начальная фаза =0; Длительность одного значения 1.

Amplitude=1; Frequency (rad/sec)=10; Faze (rad)=0; Sample time=1.

Пример результатов моделирования



Лабораторная работа №3

Исследование элементов модели системы массового обслуживания

Цель занятия: Освоить методику моделирования элементов систем массового обслуживания (СМО)

1. Общие сведения о СМО

Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования.

В любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания заявкой и собственно обслуживание заявки. Это можно изобразить в виде некоторого i -го прибора обслуживания Π_i (рис. 1), состоящего из накопителя заявок H_i в котором может одновременно находиться $l_i = \overline{L_i^H}$ заявок, где L_i^H — емкость i -го накопителя, и канала обслуживания заявок (или просто канала) K_i . На каждый элемент прибора обслуживания Π_i поступают потоки событий: в накопитель H_i — поток заявок W_i на канал K_i — поток обслуживания u_i .

Потоком событий называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Различают потоки однородных и неоднородных событий. Поток событий называется *однородным*, если он характеризуется только моментами поступления этих событий (вызывающими моментами) и задается последовательностью $\{t_n\} = \{0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq \dots\}$, где t_n — момент наступления n -го события — неотрицательное вещественное число. Однородный поток событий также может быть задан в виде последовательности промежутков времени между n -м и $(n-1)$ -м событиями $\{\tau_n\}$, которая однозначно связана с последовательностью вызывающих моментов $\tau_n = t_n - t_{n-1}, n \geq 1, t_0 = 0$, т. е. $\tau_1 = t_1$.

Потоком неоднородных событий называется последовательность $\{t_n, f_n\}$, где t_n — вызывающие моменты; f_n — набор признаков события. Например, применительно к процессу обслуживания для неоднородного потока заявок могут быть заданы принадлежность к тому или иному источнику заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т. п.

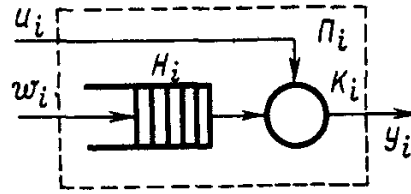


Рис. 1. Устройство обслуживания заявок

Рассмотрим поток, в котором события разделены интервалами времени τ_1, τ_2, \dots , которые вообще являются случайными величинами. Пусть интервалы τ_1, τ_2, \dots независимы между собой. Тогда поток событий называется *поток с ограниченным последствием*.

Пример потока событий приведен на рис. 2, где обозначено T_j интервал между событиями (случайная величина); T_n — время наблюдения, T_c — момент совершения события.

Интенсивность потока можно рассчитать экспериментально по формуле

$$\lambda = \frac{N}{T_n}$$

где N — число событий, произошедших за время наблюдения T_n . Если $T_j = \text{const}$ или определено какой-либо формулой $T_j = f(T_{j-1})$, то поток называется *детерминированным*. Иначе поток называется *случайным*.

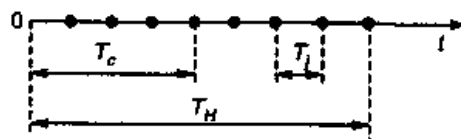


Рис.2 Графическое изображение N-схемы

Случайные потоки бывают:

- ординарными, когда вероятность одновременного появления 2-х и более событий равна нулю;
- стационарными, когда частота появления событий постоянная;
- без последствия, когда вероятность не зависит от момента предыдущих событий.

Поток событий называется *ординарным*, если вероятность того, что на интервал времени, примыкающий к моменту времени t , попадает больше одного события, пренебрежительно мала по сравнению с вероятностью того, что на этот же интервал времени попадает ровно одно событие.

Стационарным потоком событий называется поток, для которого вероятность появления того или иного числа событий на интервале времени t зависит от длины этого участка и не зависит от того, где на оси времени взят этот участок.

Интенсивность потока может быть любой неотрицательной функцией времени, имеющей размерность, обратную размерности времени. Для стационарного потока его интенсивность не зависит от времени и представляет собой постоянное значение, равное среднему числу событий, наступающих в единицу времени.

Возможные приложения. Обычно в приложениях при моделировании различных систем применительно к элементарному каналу обслуживания K_j , можно считать, что поток заявок $W_j \in W$, т. е. интервалы времени между моментами появления заявок (вызывающие моменты) на входе K_j образует подмножество неуправляемых переменных, а поток обслуживания $u_j \in U$, т. е. интервалы времени между началом и окончанием обслуживания заявки, образует подмножество управляемых переменных.

Заявки, обслуженные каналом K_j и заявки, покинувшие прибор Π_j по различным причинам необслуженными (например, из-за переполнения накопителя H_j , образуют выходной поток $y_j \in Y$, т. е. интервалы времени между моментами выхода заявок образуют подмножество выходных переменных.

Процесс функционирования прибора обслуживания Π_j можно представить как процесс изменения состояний его элементов во времени $z_j(t)$. Переход в новое состояние для Π_j означает изменение количества заявок, которые в нем находятся (в канале K_j , и в накопителе H_j). Таким образом, вектор состояний для Π_j имеет вид $\vec{z}(t) = (z_j^K, z_j^H)$, где z_j^H — состояние накопителя ($z_j^H = 0$ — накопитель пуст, $z_j^H = 1$ — в накопителе имеется одна заявка ... $z_j^H = L_j^H$ — накопитель полностью заполнен); L_j^H — емкость накопителя, измеряемая числом заявок, которые в нем могут поместиться; z_j^K — состояние канала ($z_j^K = 0$ — канал свободен, $z_j^K = 1$ — канал занят и т. д.).

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а *Q-схемы*, образуемые композицией многих элементарных приборов обслуживания (сети массового обслуживания). Если каналы K_i различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание (многоканальная *Q-схема*), а если приборы Π_j и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная *Q-схема*). Таким образом, для задания *Q-схемы* необходимо использовать оператор сопряжения R , отражающий взаимосвязь элементов структуры (каналов и накопителей) между собой.

Связи между элементами *Q-схемы* изображают в виде стрелок (линий потока,

отражающих направление движения заявок). Различают разомкнутые и замкнутые Q -схемы. В разомкнутой Q -схеме выходной поток обслуженных заявок не может снова поступить на какой-либо элемент, т. е. обратная связь отсутствует, а в замкнутых Q -схемах имеются обратные связи, по которым заявки двигаются в направлении, обратном движению вход-выход.

Собственными (внутренними) параметрами Q -схемы будут являться количество фаз обслуживания, количество каналов в каждой фазе, количество накопителей каждой фазы, емкость i -го накопителя. Следует отметить, что в теории массового обслуживания в зависимости от емкости накопителя применяют следующую терминологию для систем массового обслуживания: системы с потерями (т. е. накопитель отсутствует, а имеется только канал обслуживания), системы с ожиданием (т. е. накопитель, имеет бесконечную емкость и очередь заявок не ограничивается) и системы смешанного типа (с ограниченной емкостью накопителя). Всю совокупность собственных параметров Q -схемы обозначим как подмножество H ,

Для задания Q -схемы также необходимо описать алгоритмы ее функционирования, которые определяют набор правил поведения заявок в системе в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе и обслуживания заявок каналом, каждого элементарного обслуживающего прибора P , Q -схемы. Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов.

В зависимости от динамики приоритетов в Q -схемах различают статические и динамические приоритеты. Статические приоритета назначаются заранее и не зависят от состояний Q -схемы, т. е. они являются фиксированными в пределах решения конкретной задачи моделирования. Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций. Исходя из правил выбора заявок из накопителя на обслуживание каналом, можно выделить относительные и абсолютные приоритеты. *Относительный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель ожидает окончания обслуживания предшествующей заявки каналом K , и только после этого занимает канал. *Абсолютный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель прерывает обслуживание каналом K заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из K заявка может либо покинуть систему, либо может быть снова записана на какое-то место в H).

При рассмотрении алгоритмов функционирования приборов обслуживания (каналов, и накопителей) необходимо также задать набор правил, по которым заявки покидают P и K . Для P — либо правила переполнения, по которым заявки в зависимости от заполнения H покидают систему, либо правила ухода, связанные с истечением времени ожидания заявки в H , для K — правила выбора маршрутов или направлений ухода. Кроме того, для заявок необходимо задать правила, по которым они остаются в канале K или не допускаются до обслуживания каналом K т. е. правила блокировок канала. При этом различают блокировки K по выходу и по входу. Такие блокировки отражают наличие управляющих связей в Q -схеме, регулирующих поток заявок в зависимости от состояний Q -схемы. Весь набор возможных алгоритмов поведения заявок в Q -схеме можно представить в виде некоторого оператора алгоритмов поведения заявок A .

Таким образом, Q -схема, описывающая процесс функционирования системы массового

обслуживания любой сложности, однозначно задается в виде $Q = \langle W, U, H, Z, R, A \rangle$.

2. Моделирование основных элементов СМО с использованием пакета

Simulink

В состав модели СМО в обязательном порядке входят модели генераторов потока заявок и потока обслуженных заявок. В основу моделей генераторов положены источники последовательностей случайных чисел равномерно распределенных на интервале $(0 \dots 1)$. На основе этой последовательности можно имитировать потоки с различными законами распределения.

Пусть при моделировании некоторой системы необходимо сформировать на ЭВМ простейший поток заявок. Тогда длина интервала между $(i-1)$ -м и i -м событиями $y_i = -(1/\lambda) \ln(x_i)$.

Если при моделировании некоторой системы требуется сформировать на ЭВМ поток событий, равномерно распределенных на интервале (a, b) , то функция плотности интервалов между событиями $f(y) = 1/(b-a)$, $a \leq y \leq b$. Распределение первого интервала между началом отсчета и первым событием

$$f_1(y_1) = \lambda \left(1 - \int_0^{y_1} f(y) dy\right) = \lambda \left[1 - \int_0^{y_1} dy/(b-a)\right].$$

Интенсивность потока

$$\lambda = 1/M[y] = 1/\int_a^b y f(y) dy = 2/(a+b).$$

Тогда $f_1(y_1) = 2[1 - y_1/(b-a)]/(a+b)$.

где x_i — случайная величина, равномерно распределенная на интервале $(0, 1)$.

Для формирования на ЭВМ потока Эрланга, в котором между последовательными событиями закон распределения интервалов

$$f_k(t) = \frac{\lambda (\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, \quad t > 0.$$

плотность распределения длины первого интервала

$$f_1(y_1) = a'(t_0, y_1) e^{-a(t_0, y_1)},$$

где a — математическое ожидание числа событий на интервале $(t_0 + \Delta t)$, $a(t_0, y_1) = -\ln(x_1)$.

Примерный состав модели генератора представлен на рис.2

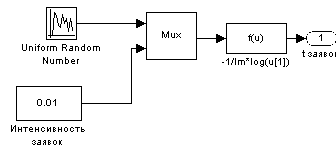


Рис.2 . Генератор

Функция, реализованная в генераторе, соответствует заданному закону распределения случайных величин. Генератор обслуженных заявок имеет аналогичную структуру.

Основной задачей накопителя является сравнение времени обслуживания заявки и времени поступления новой заявки. Если предыдущая заявка не была обслужена, формируется признак наличия задержанной заявки. На рис.3 показан примерный вид накопителя.

Вариант объединения накопителя и двух генераторов показан на рис.4.

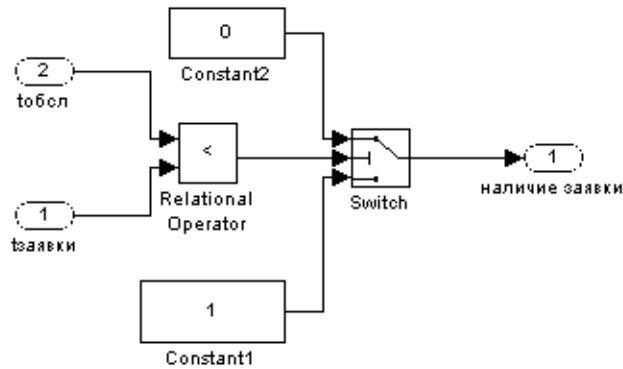


Рис. 3. Накопитель

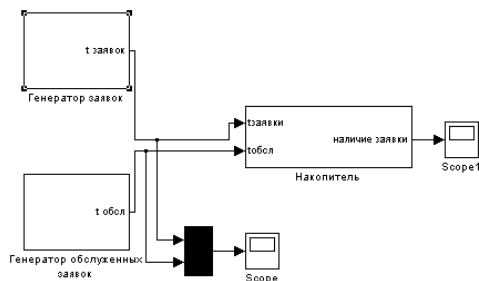


Рис. Пример соединения накопителя и генераторов

Лабораторная работа №4.

Исследование модели системы массового обслуживания

Цель занятия: Освоить методику моделирования элементов систем массового обслуживания (СМО)

В зависимости от характера источника заявок различают разомкнутые и замкнутые системы массового обслуживания СМО. На практическом занятии будут рассмотрены только разомкнутые СМО (рис.1). В зависимости от числа мест в очереди различают СМО с отказами и без отказов. В СМО с отказами число мест в очереди конечно и вследствие вероятностного характера, как входящего потока, так и процессов обслуживания, существует ненулевая вероятность того, что поступившая на вход СМО заявка застанет все каналы занятыми обслуживанием и все места в очереди занятыми заявками, ожидающими обслуживания, т. е. она получит отказ. В СМО без отказов заявка либо сразу назначается на обслуживание, если в момент ее поступления свободен хотя бы один канал, либо безусловно принимается в очередь.

1. Построение сетевых моделей одноканальных систем массового обслуживания

Кратко рассмотрим методику составления сетевых моделей на примере одноканальной СМО (в совокупности с генератором заявок). Следует отметить, что процесс разработки сетевых моделей в общем случае является неформализованным. Проинтерпретируем СМО в терминах транзакций и ресурсов. Транзакции – это активные подвижные элементы системы, а ресурсы – неактивные. Транзакциями в СМО являются заявки, а ресурсом является канал обслуживания заявок. Функционирование СМО описывается как взаимодействие транзакций и ресурсов.

При переходе от системы транзакций и ресурсов к сетевой модели можно пользоваться следующими правилами. Каждый ресурс представляется позицией, причем маркировка этой позиции (простыми метками) определяет состояние ресурса. В одноканальной СМО имеется один ресурс, имеющий два состояния: "занят" и "свободен", причем начальное состояние ресурса – "свободен". Поставим этому ресурсу в соответствие позицию R сетевой модели. Маркировка $M(R) = 1$ будет свидетельствовать о незанятости ресурса, а $M(R) = 0$ – о его занятости. Маркировка $M(R) \geq 2$ является запрещенной. Переход ресурса из одного состояния в другое представляется изменением маркировки позиции R : добавление метки соответствует освобождению ресурса, а изъятие – его занятию.

В системе транзакций и ресурсов каждая транзакция представляется простой меткой или меткой с атрибутами в зависимости от того, несет она информацию или нет. Будем считать, что в нашей одноканальной СМО каждая транзакция несет информацию и, следовательно, будет представляться меткой с атрибутами.

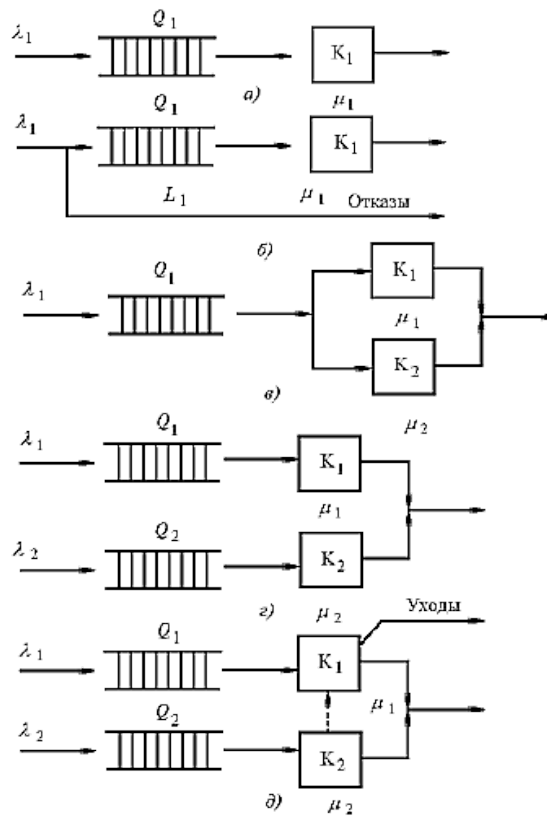


Рис. 1. Системы массового обслуживания:
а, б – одноканальные;
в, г, д – двухканальные

Транзакция в процессе ее обработки может находиться в нескольких состояниях, определяющих степень ее готовности. Для одноканальной СМО можно выделить следующие состояния транзакции:

- 0 – "готова для планирования";
- 1 – "планируется";
- 2 – "ожидание обслуживания";
- 3 – "заняла канал обслуживания";
- 4 – "обрабатывается";
- 5 – "обработана";
- 6 – "покинула канал обслуживания".

Состояния транзакции можно разделить на следующие классы:

- состояния временной активности (1, 4), связанные с процессами обработки транзакции, а также с планированием транзакции в генераторе транзакций. В этих состояниях транзакция находится во временной задержке перехода;
- состояния ожидания какого-либо ресурса (2). В этих состояниях транзакция находится в позициях, в которых нередко образуются очереди;
- состояния без ожидания (0, 3, 5, 6). В этих состояниях транзакция только "заявляет" о необходимости выполнения какого-либо события.

Каждому состоянию транзакции поставим в соответствие позицию или временную задержку перехода сетевой модели.

Переход транзакции из одного состояния в другое представляется как перемещение метки из одной позиции в другую, из позиции во временную задержку перехода, а также из временной задержки перехода в позицию.

В дальнейшем в моделируемой системе определяется множество событий, которые приводят к изменению состояния транзакций и ресурсов. В одноканальной СМО с генерацией заявок можно выделить следующие события:

1. "Приход новой заявки". При возникновении этого события транзакция переходит из состояния 1 в состояние 2. Создается копия транзакции, которая переводится в состояние 0.

2. "Планирование следующей заявки". Транзакция из состояния 0 переходит в состояние 1.

3. "Занятие канала обслуживания". Транзакция из состояния 2 переходит в состояние 3, а ресурс из состояния "свободен" переходит в состояние "занят".

4. "Начало обработки заявки". Транзакция из состояния 3 переводится в состояние 4.

5. "Конец обработки заявки". Транзакция из состояния 4 переводится в состояние 5.

6. "Освобождение канала обслуживания". Транзакция из состояния 5 переходит в состояние 6, а ресурс из состояния "занят" переходит в состояние "свободен".

7. "Уничтожение заявки". Транзакция покидает состояние 6 и уничтожается.

2. Моделирование очереди заявок

Очередью называется элемент исследуемого объекта, с помощью которого моделируются процессы ожидания начала обработки требований обслуживающими устройствами. Возникновение этих процессов происходит ввиду того, что в момент поступления требования устройство может находиться в состояниях "занято" либо "выключено". Кроме того, ожидание начала обработки может возникнуть из-за того, что приоритет поступившего требования меньше, чем приоритет обрабатываемого. По любой из вышеперечисленных причин, поступающие требования устанавливаются в очередь к устройству.

На входе устройства может формироваться произвольное количество очередей. При этом количество очередей не может превышать общего числа внесистемных и внутрисистемных источников, заданных в предложении *входящий поток*. В зависимости от количества этих источников и правил функционирования элементов, имитирующих процессы ожидания, для описания очередей используется одна из следующих конструкций:

очередь: общая / отдельная / смешанная;

С помощью первой конструкции формируется одна очередь, в которую могут устанавливаться все требования, поступающие на вход устройства. Остальные две конструкции позволяют формировать множество очередей перед устройством. Независимо от количества очередей, установленных на входе устройства, дисциплины их формирования определяются предложением *механизм обслуживания*.

Общая очередь

Общей очередью называется такая очередь, в которую могут устанавливаться требования, поступающие от всех внесистемных и внутрисистемных источников, указанных в предложении *входящий поток*. Задание общей очереди осуществляется с помощью конструкции *общая*. Формирование очереди производится независимо от способа передачи требований на вход устройства. Конкретное место, занимаемое в очереди каждым требованием, определяется в зависимости от его приоритета и механизма обслуживания. При этом основополагающим правилом является механизм обслуживания, устанавливающий дисциплину формирования очереди.

Пример модели обслуживающего устройства ($n,1$) с общей очередью представлен на рис. 1.

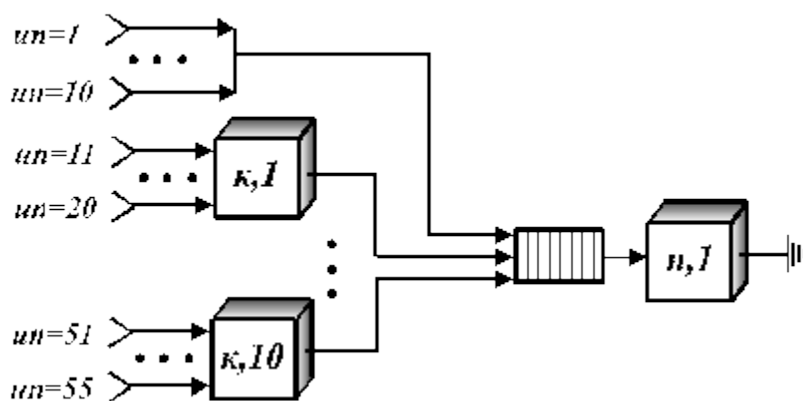


Рис. 1. Схема модели устройства с общей очередью

На вход устройства поступают требования от 10 внесистемных источников с индексами от 1 до 10 включительно и от 45 внутрисистемных источников. Интенсивность первых пяти источников требований равна 0.1, а следующих пяти источников - 0.2. Как показано на рисунке, на вход устройства $(n, 1)$ с устройства $(κ, 1)$ поступают требования от внутрисистемных источников с индексами от 11 до 20 включительно, а с устройства $(κ, 10)$ с индексами от 51 до 55 включительно. На устройствах $(κ, 2) - (κ, 9)$ обрабатываются требования, генерируемые внесистемными источниками с индексами от 21 до 50 включительно. Описание обслуживающего устройства с общей очередью можно представить следующим образом:

$у(n, 1); вп: (un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11-20 \text{ у}(κ, 1)), \dots (un=51-55 \text{ у}(κ, 10));$
од: общая; мо: впп;

С применением средств интегрированного описания, позволяющих использовать фиктивные связи, модель устройства $(n, 1)$ можно представить в виде:

$у(n, 1); вп: (un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11-55 \text{ у}(κ, 1)2-10); од: общая;$
мо: впп;

Во втором случае при наличии одного внутрисистемного входа во входящем потоке конструкция *од:общая*; также позволяет сформировать общую очередь на входе $(n, 1)$.

Отдельные очереди

Отдельными называются такие очереди, которые формируются автоматически для каждого внесистемного и внутрисистемного источника, заданного во входящем потоке.

При наличии нескольких очередей на входе устройства возникает вопрос об их приоритетах, который может решаться двумя способами: по правилу умолчания либо путем явного описания очередей в порядке убывания их приоритетов. Первый способ основывается на механизме обслуживания и порядке следования описаний внесистемных и внутрисистемных источников в предложении *входящий поток*. При этом используется конструкция *очередь: отдельная*. Установка приоритетов очередям осуществляется следующим образом. Вначале формируются очереди для всех внесистемных источников. Приоритеты этих очередей определяются порядком следования описаний внесистемных источников во входящем потоке. Затем строятся очереди для внутрисистемных источников. Приоритеты этих очередей всегда ниже приоритетов очередей, формируемых внесистемными источниками. Установка приоритетов таким очередям также осуществляется в порядке очередности появления соответствующих описаний внутрисистемных источников во входящем потоке. В процессе моделирования приоритеты очередей, установленные по правилу умолчания, могут изменяться. Подробно процедуры модификации этих приоритетов освещены при описании механизма обслуживания. Задание приоритетов очередям с использованием правила умолчания рассмотрим на примере модели, представленной на рис. 2.

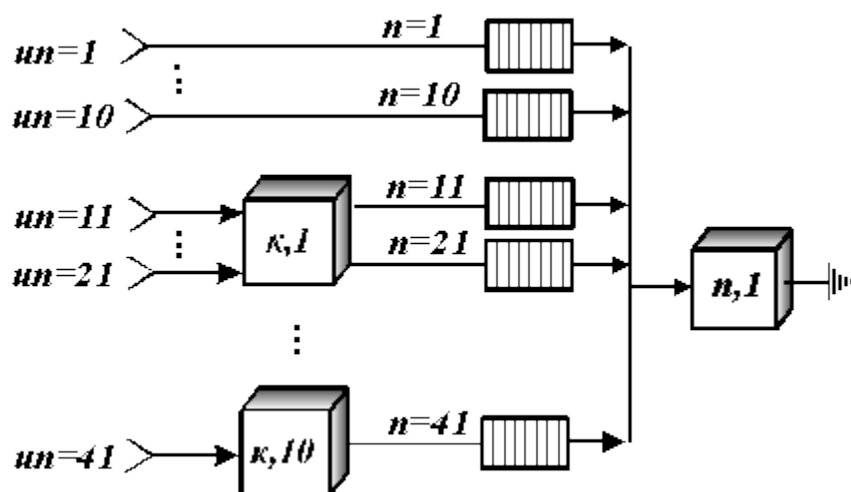


Рис. 2. Схема модели устройства с отдельными очередями

Как показано на схеме модели, на входе устройства (n, I) формируются отдельные очереди, образуемые десятью внесистемными потоками с индексами от 1 до 10 включительно и тридцатью внутрисистемными потоками с индексами от 11 до 41 включительно, поступающими с устройств (κ, I) - $(\kappa, 10)$. Описание этого фрагмента модели, иллюстрирующего формирование отдельных очередей, можно представить следующим образом:

$y(n, I); vn:(un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11 \text{ у}(\kappa, 1)), (un=12 \text{ у}(\kappa, 1)), \dots (un=21 \text{ у}(\kappa, 1)), \dots (un=41 \text{ у}(\kappa, 10));$
од: отдельная;

Согласно приведенному описанию на входе устройства (n, I) формируются отдельные очереди. Каждый внутрисистемный источник описывается независимо, что позволяет сформировать сорок одну очередь.

Наиболее простой способ модификации умалчиваемых значений приоритетов отдельных очередей состоит в изменении последовательности описания источников требований во входящем потоке. Предположим, что в рассматриваемом примере очередь на входе (n, I) , формируемая 41 источником требований $(un=41 \text{ у}(\kappa, 1))$, должна быть наиболее приоритетной среди очередей, образующихся из требований, поступающих от внутрисистемных источников. Для реализации такой модели описание входящего потока должно быть выполнено следующим образом:

$y(n, I); vn:(un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=41 \text{ у}(\kappa, 10)),$
 $(un=11 \text{ у}(\kappa, 1)), (un=12 \text{ у}(\kappa, 1)), \dots, (un=40 \text{ у}(\kappa, 9));$
од: отдельная;

Приоритеты очередей, формируемых внесистемными источниками, также определяются согласно очередности появления соответствующих описаний. Поэтому для того, чтобы очередь, формируемая десятым внесистемным источником была наиболее приоритетной среди остальных очередей, входящий поток необходимо описать следующим образом:

$y(n, I); vn:(un=10 \text{ э}(0.2)), (un=1-5 \text{ э}(0.1)), (un=6-9 \text{ э}(0.2)), (un=41 \text{ у}(\kappa, 10)), \dots;$

При использовании правила умолчания для задания приоритетов очередям применение средств интегрированного описания внесистемных и внутрисистемных источников имеет одну особенность. Она состоит в том, что независимо от степени интеграции описания внесистемного входа для каждого из перечисленных в нем источников строятся отдельные очереди, а для внутрисистемного входа всегда создается одна очередь.

В тех случаях, когда невозможно или нецелесообразно использовать неявный способ задания приоритетов очередям, можно применить второй способ установки приоритетов. Он предполагает задание приоритетов очередям путем последовательного описания соответствующих входов в предложении *очередь*:

отдельная

((<вход 1>) [<атрибуты очереди>])
[,(<вход 2>) [<атрибуты очереди>]]
[...,((<вход N>)[<атрибуты очереди>])]]

Представленная синтаксическая конструкция позволяет первой очереди, задаваемой с помощью конструкции <вход 1>, установить наивысший приоритет, а последней очереди, <вход N> - наименьший. Например, конструкция **од: отдельная** ((*un*=4)), ((*un*=11-15 *y*(*κ*,1))), ((*un*=8-10)); устанавливает убывающие приоритеты очередям, формируемым различными типами источников в следующем порядке:

- 1 очередь - ((*un*=4));
- 2 очередь - ((*un*=11-15 *y*(*κ*,1)));
- 3 очередь - ((*un*=8));
- 4 очередь - ((*un*=9));
- 5 очередь - ((*un*=10));

Из приведенного примера видно, что для каждого внесистемного источника всегда формируется отдельная очередь, хотя в тексте модели задание третьей, четвертой и пятой очередями осуществляется с помощью интегрированного входа ((*un*=8-10)).

При построении отдельных очередей не обязательно описывать все источники требований, заданные во входящем потоке. В этом случае алгоритм построения очередей и установки им необходимых приоритетов будет основываться на использовании двух указанных выше способов. Наиболее приоритетными будут очереди, формируемые на основе их явного описания. Для неописанных источников очереди создаются по правилу умолчания. Предположим, что описание входящего потока и очереди для устройства (*n*,1) представлено следующим образом:

вп: ((*un*=11-21 *y*(*κ*,1)), (*un*=22-24 *y*(*κ*,2)), (*un*=25-26 *y*(*κ*,3)),
((*un*=27-28 *y*(*κ*,4)), (*un*=29-30 *y*(*κ*,5)), (*un*=31-32 *y*(*κ*,6)),
((*un*=33-34 *y*(*κ*,7)), (*un*=35-36 *y*(*κ*,8)), (*un*=37-40 *y*(*κ*,9)),
((*un*=41 *y*(*κ*,10)),(*un*=1-10 э(0.01));
од: **отдельная** ((*un*=25-26 *y*(*κ*,3))), ((*un*=1)), ((*un*=11-21 *y*(*κ*,1)));

Согласно этому описанию наивысшим приоритетом будет обладать очередь, формируемая из требований, поступающих с устройства (*κ*,3). Второй по порядку приоритет будет иметь очередь, образуемая из требований потока с индексом 1. Следующим по порядку приоритетом будет обладать очередь, формируемая из требований потоков с индексами от 11 до 21 включительно. Приоритеты остальным очередям задаются по правилу умолчания.

Смешанные очереди

Конструкция **смешанная** предназначена для формирования очередей, в которые могут устанавливаться требования, поступающие от различных типов источников. С помощью этой конструкции пользователю предоставляется возможность задавать различные правила объединения потоков для формирования очереди. Вслед за конструкцией **смешанная** необходимо указать, из каких источников должна формироваться первая очередь, затем вторая и т.д. Задание правила формирования очереди производится путем перечисления входов, образующих эту очередь. Далее может следовать описание атрибутов очереди, определяющих некоторые особенности ее функционирования. Синтаксис смешанной очереди представляется следующим образом:

смешанная

((<вход>)[,<вход>][...,<вход>] [<атрибуты очереди>]]]
[,(<вход>)[,<вход>][...,<вход>] [<атрибуты очереди>]]]
[...,((<вход>)[,<вход>][...,<вход>])] [<атрибуты очереди>]]]

Каждая из очередей, формируемых с использованием конструкции **смешанная**, может образовываться из внесистемных и внутрисистемных источников. Ниже приведен пример модели, в которой на входе устройства (*n*,1) имеются две очереди (рис. 3).

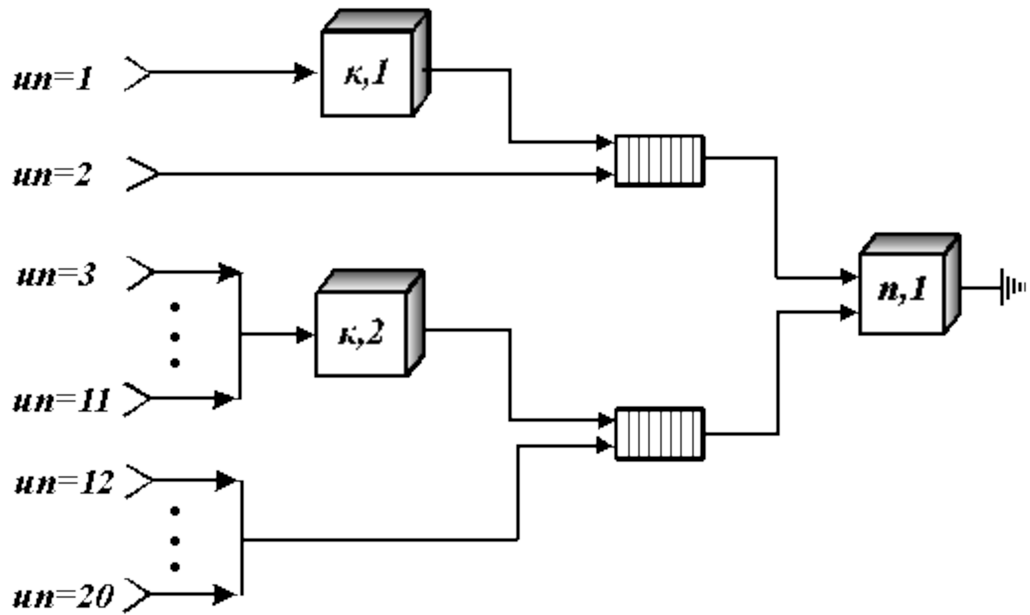


Рис. 3. Схема модели устройства со смешанными очередями

Первая из них формируется внутрисистемным потоком с индексом 1, поступающим с устройства $(\kappa, 1)$, и внесистемным потоком с индексом 2. Вторая очередь также образуется из различных типов источников. При этом внутрисистемные потоки с индексами 3 - 11 поступают с устройства $(\kappa, 2)$. Кроме того, в очередь могут устанавливаться требования, поступающие от девяти внесистемных источников с номерами 12 - 20. Описание фрагмента модели, иллюстрирующего задание смешанных типов очередей, представляется следующим образом:

$y(n, 1)$; $вп$: $(un=1 y(\kappa, 1))$, $(un=2 \varepsilon(0.1))$, $(un=3-11 y(\kappa, 2))$, $(un=12-20 \varepsilon(0.2))$;
од: смешанная $((un=1 y(\kappa, 1))$, $(un=2))$, $((un=3-11 y(\kappa, 2))$, $(un=12-20))$;

Если в описании смешанной очереди перечислены не все входы, то при построении очередей по умолчанию для каждого из них создается отдельная очередь. Предположим, что в рассмотренной выше модели для девятнадцатого и двадцатого потоков необходимо построить отдельные очереди. Для этого достаточно при описании правила формирования второй очереди опустить в перечислении девятнадцатый и двадцатый потоки:

$y(n, 1)$; $вп$: $(un=1 y(\kappa, 1))$, $(un=2 \varepsilon(0.1))$, $(un=3-11 y(\kappa, 2))$, $(un=12-20 \varepsilon(0.2))$;
од: смешанная $((un=1 y(\kappa, 1))$, $(un=2))$, $((un=3-11 y(\kappa, 2))$, $(un=12-18))$;

Этой же цели можно достигнуть путем прямого задания правила формирования третьей и четвертой очередей для 19-го и 20-го потоков:

$y(n, 1)$; $вп$: $(un=1 y(\kappa, 1))$, $(un=2 \varepsilon(0.1))$, $(un=3-11 y(\kappa, 2))$, $(un=12-20 \varepsilon(0.2))$;
од: смешанная $((un=1 y(\kappa, 1))$, $(un=2))$, $((un=3-11 y(\kappa, 2))$, $(un=12-18))$, $((un=19))$, $((un=20))$;

Если необходимо, чтобы 19 и 20-ый потоки образовывали одну очередь, то описание очереди должно быть представлено следующим образом:

од: смешанная $((un=1 y(\kappa, 1))$, $(un=2))$, $((un=3-11 y(\kappa, 2))$, $(un=12-18))$, $((un=19))$, $((un=20))$;

Как и в случае с отдельным типом очереди, при наличии более одной очереди на входе устройства, возникает вопрос об их приоритетах. Приоритет задается согласно очередности описания правила формирования очереди. Первая очередь, встретившаяся в описании, имеет наивысший приоритет. Приоритет остальных очередей последовательно убывает. Если при описании правил формирования смешанной очереди указаны не все входы, перечисленные во входящем потоке, то для них по умолчанию также создаются отдельные очереди. При этом сначала строятся очереди для внесистемных, а затем - для внутрисистемных потоков.

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

	Авторы,	Заглавие	Издательство, год	Адрес	
Л1.1	Блинков, Ю. В.	Основы теории информационных процессов и систем: учебное пособие	Пенза: Пензенский государственный университет архитектуры и строительства, ЭБС АСВ, 2011	http://www.iprbookshop.ru/23103.html	
Л1.2	Белов, П. С.	Математическое моделирование технологических процессов: учебное пособие (конспект лекций)	Егорьевск: Егорьевский технологический институт (филиал) Московского государственного технологического университета «СТАНКИН», 2016	http://www.iprbookshop.ru/43395.html	
6.1.2. Дополнительная литература					
	Авторы,	Заглавие	Издательство, год	Адрес	
Л2.1	Казиев В. М.	Введение в анализ, синтез и моделирование систем	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/52188.html	
Л2.2	Плохотников, К. Э.	Методы разработки математических моделей и вычислительный	Москва: СОЛОН-ПРЕСС, 2017	http://www.iprbookshop.ru/64926.html	
6.1.3. Методические разработки					
	Авторы,	Заглавие	Издательство, год	Адрес	
Л3.1	Татарникова, Т. М.	Моделирование систем: методические указания к выполнению лабораторных работ	Санкт-Петербург: Российский государственный гидрометеорологический университет, 2008	http://www.iprbookshop.ru/12503.html	
Л3.2	Шевцова, Ю. В.	Математические модели и методы исследования операций: сборник задач	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2009	http://www.iprbookshop.ru/54766.html	
Л3.3	Сёмина, В. В.	Моделирование систем: методические указания для проведения лабораторных работ по дисциплине «моделирование	Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2016	http://www.iprbookshop.ru/64869.html	
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"					
Э1	Душин В.К.	Теоретические основы информационных процессов и систем [Электронный ресурс]: учебник/ В.К.— Электрон. текстовые данные.— М.: Дашков и К, 2014.— 348 с.— Режим доступа: http://www.iprbookshop.ru/24764 .— ЭБС «IPRbooks», по паролю			
Э2	Шатрова Г.В.	Методы исследования и моделирования информационных процессов и технологий [Электронный ресурс]: учебное пособие/ Шатрова Г.В., Топчиев И.Н.— Электрон. текстовые данные.— Ставрополь: СКАУ, 2016.— 180 с			

ЭЗ	Лубенец Ю.В. Экономико-математические методы и модели [Электронный ресурс]: учебное пособие/ Электрон. текстовые данные.— Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013.	Лубенец Ю.В.
----	--	--------------



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Анализ и синтез информационных систем» для
студентов направления подготовки
090402 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Анализ и синтез информационных систем» содержат задания для студентов, необходимые для практических занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 090402 Информационные системы и технологии, направленность (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1

Лабораторная работа 2

Лабораторная работа 3

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины - дать представление о перспективах развития методов анализа и синтеза информационных систем, изучить методы анализа и синтеза информационных систем и методику их применения, сформировать умения использовать методы анализа и синтеза информационных систем на практике.

В результате освоения данной дисциплины формируется следующая компетенция у обучающегося:

ОПК-5.2: Использует современное программное и аппаратное обеспечение информационных и автоматизированных систем

УК-1.3: Использует методологию системного и критического анализа проблемных ситуаций, методики постановки цели, определения способов ее достижения

Изучив данный курс, студент должен:

Знать:

принципы системного представления основных этапов проектирования информационных систем, основанного на объектном подходе.

Уметь:

использовать промышленные стандартизированные решения, опирающиеся на современные технологии;

проектировать информационные системы от этапа постановки задачи до программной реализации.

Владеть:

методами анализа информационных ресурсов; разработки различных моделей данных; конструирования программных модулей; анализа проектных решений.

Реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа №1

Оценка надежности электронного блока (модуля) ИС

1. Цель работы

Изучение методики расчета показателей надежности электронного модуля при экспоненциальном законе распределения отказов элементов.

2. Теоретическая часть

2.1. Основные понятия и термины

Система – объект, представляющий совокупность элементов, взаимодействующих в процессе выполнения определенного круга задач и взаимосвязанных функционально. В лабораторной работе под системой понимается электронный модуль, предназначенный для решения определенной задачи.

Элемент системы – объект, представляющий собой простейшую часть системы, например резистор, конденсатор, трансформатор и т.п.

Надежность – свойство объекта сохранять во времени в установленных пределах значения всех параметров.

Работоспособное состояние – состояние объекта, при котором значения всех параметров, характеризующих способность выполнять заданные функции, соответствуют требованиям нормативно-технической документации.

Неработоспособное состояние – состояние объекта, при котором значение хотя бы одного параметра, характеризующего способность объекта выполнять заданные функции, не соответствует требованиям нормативно-технической документации.

Отказ – событие, заключающееся в нарушении работоспособного состояния объекта.

Внезапный отказ характеризуется скачкообразным изменением значений одного или нескольких основных параметров объекта.

Наработка – продолжительность или объем работы объекта.

Наработка до отказа – наработка объекта от начала его эксплуатации до возникновения первого отказа.

Невосстанавливаемый объект – объект, для которого в рассматриваемой ситуации проведение восстановления работоспособного состояния не предусмотрено нормативно-технической документацией.

Технология изготовления современных элементов аппаратуры столь сложна, что не всегда удается проследить за скрытыми дефектами производства, которые должны выявляться на стадии тренировки и приработки аппаратуры. В результате в сферу эксплуатации могут проникать следующие дефектные элементы: резистор с недостаточно прочным креплением токоотвода; токопроводящий слой печатного монтажа, у которого толщина либо чрезвычайно малая, либо чрезмерно большая; интегральная схема, у которой соединение вывода с печатным монтажом недостаточно прочное и т.д. В процессе эксплуатации могут создаваться условия, при которых скрытый дефект приводит

к отказу изделия (пиковые нагрузки, вибрация, температурный скачок, помехи и т.д.). При большом уровне случайных неблагоприятных воздействий внезапный отказ может произойти даже при отсутствии скрытых дефектов.

Необходимо отличать логическую схему соединения элементов расчета надежности от электрической схемы соединения радиоэлектронных элементов.

При расчете надежности объекта строится логическая схема соединения его элементов.

В качестве простого примера рассмотрим два параллельно включенных диода, выполняющих функцию выпрямителя (рис.1).

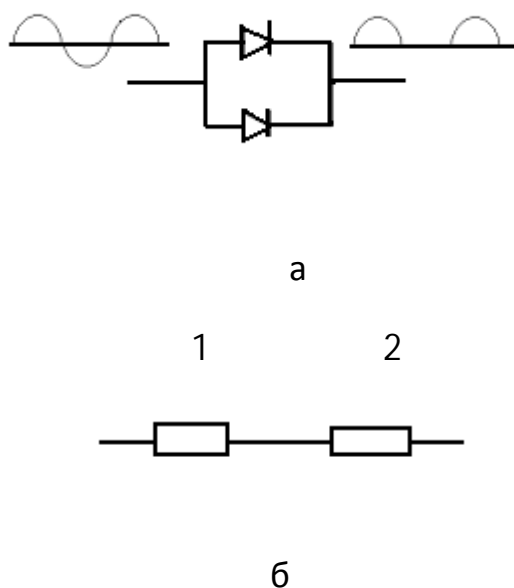


Рис.1.Схемы соединения элементов:
а-электрическая схема; б-логическая схема

В принципе эту функцию может выполнить один диод; они запараллелены с целью облегчения режима их работы. С точки зрения отказа «короткое замыкание» их логическая схема – последовательное соединение.

Таким образом, если модуль выходит из строя при отказе любого из его N элементов, то логическая схема расчета надежности состоит из N последовательно соединенных элементов. Надежность таких объектов исследуется в этой лабораторной работе.

2.2. Показатели надежности невосстанавливаемых объектов

Показатель надежности – характеристика, определяющая одно или несколько свойств, составляющих надежность объекта.

Вероятность безотказной работы за наработку t_3 есть вероятность того, что объект проработает безотказно в течение заданного времени t_3 , начав работать в момент времени $t=0$:

$$P(0, t_3) = P(t_3) = P(T >= t_3), \quad (1)$$

где t_3 – заданная наработка; T – случайная величина, представляющая наработку объекта до отказа.

Другими словами, $P(t_3)$ является монотонно убывающей функцией, причем очевидно $P(0)=1$ и $P(t_3=\infty) = 0$, так как любой объект, работоспособный в момент включения, со временем откажет.

Вероятность отказа есть вероятность того, что объект откажет в течение заданной наработки от 0 до t_3 :

$$Q(t_3) = 1 - P(t_3) = P(T < t_3). \quad (2)$$

Очевидно, что $Q(0) = 0$ и $Q(t_3=\infty) = 1$.

Вероятность безотказной работы объекта на интервале времени от t_1 до t_2 можно определить из соотношения

$$P(t_1, t_2) = P(t_2)/P(t_1). \quad (3)$$

Плотность вероятности отказа есть производная от вероятности отказа невозстанавливаемого объекта

$$f(t) = \frac{dQ(t)}{dt} = -\frac{dP(t)}{dt}. \quad (4)$$

Одним из основных расчетных показателей надежности является *интенсивность отказов*:

$$\lambda(t) = \frac{f(t)}{P(t)} = -\frac{dP(t)}{dt} / P(t). \quad (5)$$

Если этот показатель известен, то другие показатели легко рассчитать:

$$P(t_3) = \exp \left\{ -\int_0^{t_3} \lambda(t) dt \right\}; \quad (6)$$

$$Q(t_3) = 1 - \exp \left\{ -\int_0^{t_3} \lambda(t) dt \right\}; \quad (7)$$

$$P(t_1, t_2) = \exp \left\{ -\int_{t_1}^{t_2} \lambda(t) dt \right\}. \quad (8)$$

На участке нормальной эксплуатации преобладают случайные внезапные отказы, поэтому обычно принимают интенсивность отказов $\lambda(t) = \Lambda = const$. В этом случае говорят, что используется экспоненциальный закон надежности, для которого верны следующие расчетные соотношения:

$$P(t_3) = e^{-\Lambda t_3}; \quad (9)$$

$$Q(t_3) = 1 - e^{-\Lambda t_3}; \quad (10)$$

$$f(t_3) = \lambda e^{-\Lambda t_3}; \quad (11)$$

$$P(t, t_2) = e^{-\Lambda(t_2 - t_1)}. \quad (12)$$

В качестве показателя надежности невосстанавливаемого объекта широко используется *средняя наработка до отказа* – математическое ожидание наработки объекта до отказа.

$$T_{cp} = \int_0^{\infty} t f(t) dt = \int_0^{\infty} P(t) dt. \quad (13)$$

Если $\lambda(t) = \Lambda = const$, то средняя наработка до отказа

$$T_{cp} = \int_0^{\infty} e^{-\Lambda t} dt = \frac{1}{\Lambda}, \quad (14)$$

т.е. при экспоненциальном законе надежности средняя наработка до отказа обратно пропорциональна интенсивности отказов.

Тогда

$$P(t) = e^{-\frac{t}{T_{cp}}} \text{ и } f(t) = \frac{1}{T_{cp}} e^{-\frac{t}{T_{cp}}}. \quad (15)$$

Если $t = T_{cp}$, то

$$P(T_{cp}) = e^{-1} \approx 0,37,$$

т.е. в этом случае под средней наработкой до отказа можно понимать такую наработку, по истечении которой из множества одинаковых объектов в среднем должны остаться работоспособными 37 %.

2.3. Методика ориентировочного расчета надежности

Ориентировочный расчет обычно сопровождается разработку эскизного проекта, когда еще нет полных данных о режимах и условиях работы отдельных элементов разрабатываемого изделия, но уже известна структура, т.е. состав элементов и связи между ними.

Расчет производится обычно при следующих допущениях:

- а) отказы элементов являются событиями случайными и независимыми;
- б) выявляются элементы, отказ которых не приводит к отказу системы. Эти элементы при расчёте надёжности не учитываются;
- в) вероятность безотказной работы элементов изменяется от времени по экспоненциальному закону;
- г) однотипные элементы объединяются в группы. Пусть число элементов j -го типа будет N_j , а общее число групп элементов M ;
- д) учёт влияния условий работы производится приближённо.

Составляется таблица (табл. 1), в колонки которой вносится наименование (тип) элементов, их число N_j , базовое значение λ_{0j} интенсивности отказов элементов j -й группы и произведение $N_j * \lambda_{0j}$. Затем рассчитывается интенсивность отказов всего устройства:

$$\Lambda = K_{\text{уэ}} \cdot \sum_{j=1}^M N_j \lambda_{0j}, \quad (16)$$

$K_{\text{уэ}}$ – поправочный коэффициент, учитывающий влияние условий эксплуатации системы (табл. 2).

Значения интенсивностей отказов λ_{0j} элементов определяются по справочникам.

Основные количественные характеристики надёжности системы рассчитываются по формулам (9)-(15).

Таблица 1

Наименование групп элементов	Количество элементов в группе N_j	Базовая интенсивность отказов элементов в группе λ_{0j} , 1/ч	$N_j * \lambda_{0j}$, 1/ч
Микросхемы	20	0.000001	0.00002
Транзисторы	10	0.000003	0.00003
Резисторы	30	0.000001	0.00003
			$\Lambda = \sum N_j * \lambda_{0j}$

Таблица 2

Условия эксплуатации	Значения поправочного коэффициента $K_{\text{уэ}}$
Земля, лабораторные условия	1
Стационарная наземная	10
Корабельная	17
Передвижные платформы	30
Высокогорная	80
Дозвуковая авиационная	120
Сверхзвуковая авиационная	350

3. Порядок выполнения лабораторной работы

1. Запустить на выполнение программу pad.exe (рис.2) и в появившемся диалоговом окне *Расчет Надежности* ввести номер варианта исследуемого модуля. Принципиальная схема электронного блока отображается на экране. Нажать на кнопку «*Описание*» и вывести на экран текстовое описание назначения и принципа действия исследуемого устройства.
2. Определить группы элементов, которые используются в принципиальной схеме, и подсчитать количество элементов, содержащихся в каждой группе. Выбор названия группы элементов производится из выпадающего списка *Выберите элемент*, число элементов указывается в поле *Количество в схеме*. После каждого выбора группы элементов нажать кнопку «*Ввести данные*». В результате введенные данные пересылаются в Excel.
3. После завершения процесса определения всех групп элементов принципиальной схемы необходимо перейти в среду Excel (кнопка «*Открыть EXCEL*») для выполнения расчетов количественных характеристик надежности.
4. Открыть автоматизированную справочно-информационную систему по расчету надежности (программа ASSRN.exe). Для каждой группы элементов определить базовую интенсивность отказов и ввести ее в колонку *Базовая интенсивность отказов элементов в группе* таблицы расчетов надежности.
5. Заполнить таблицу расчета надежности (см. табл.1). Рассчитать групповую интенсивность отказов. С этой целью для каждой группы элементов с помощью математических функций Excel найти произведение количества элементов данной группы и базовой интенсивности отказов. Рассчитать интенсивность отказов всего устройства путем суммирования всех произведений с помощью функции суммы.

Для того чтобы получить сумму интенсивностей отказов необходимо выполнить следующие действия:

- а) щелкнуть кнопкой мыши по свободной ячейке, в которой вы хотите получить результат суммирования;
 - б) на панели инструментов выбрать знак «*=*»;
 - в) в открывшейся форме выбрать из раскрывшегося списка операцию СУММ;
 - г) в поле значений для «*числа1*» выделить диапазон чисел, которые необходимо суммировать, т.е числа в колонке *Интенсивность отказов устройства*;
 - д) нажать кнопку «*ОК*».Результат отображается в выбранной свободной ячейке.
6. Результат, полученный в п. 5, умножить на поправочный коэффициент, учитывающий условия эксплуатации (см. табл. 2).
 7. Используя формулы (9)-(14) рассчитать основные показатели надежности электронного блока.
 8. Построить соответствующие графики. Используйте *Мастер Диаграмм* Excel. Для этого выделите мышью значения, по которым необходимо построить график. На панели инструментов выберите пункт ВСТАВКА и подпункт ДИАГРАММА. Тип диаграммы – график. После выбора дополнительных параметров графика нажмите кнопку «*Готово*».

9. Составить отчёт по лабораторной работе. Он должен содержать схему электронного блока и её краткое описание, таблицу расчета надёжности, количественные характеристики надёжности блока и соответствующие графики, анализ влияния условий эксплуатации на надёжность электронного блока.

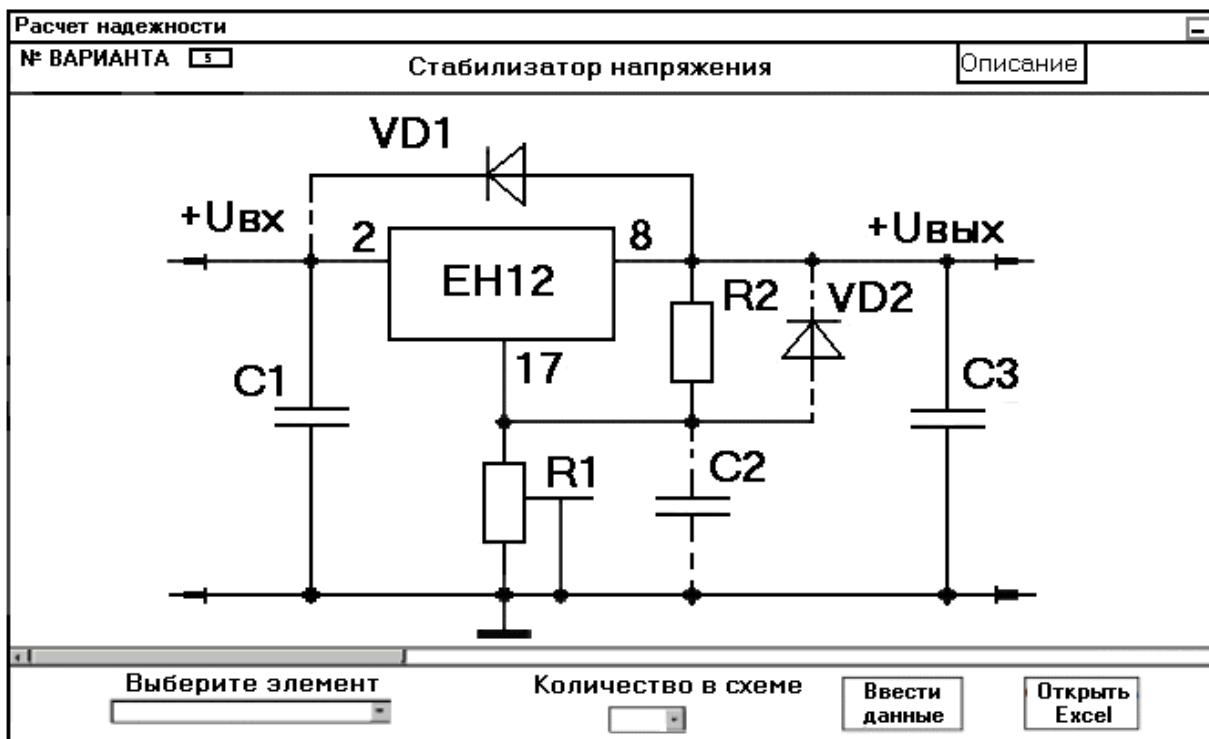


Рис.2. Окно программы nad.exe

4. Контрольные вопросы

1. Перечислите и дайте определение основных понятий и терминов, используемых в лабораторной работе.
2. Чем отличается логическая схема соединения элементов расчета надежности от электрической схемы?
3. Какие вы знаете показатели надежности невосстанавливаемых объектов?
4. Как найти среднюю наработку до отказа по графику вероятности безотказной работы системы?
5. Поясните методику ориентировочного расчета надежности.
6. Как найти количественные характеристики надежности системы, если известна средняя наработка до отказа ($T_{ср}$)?
7. Какие факторы не учитываются при ориентировочном расчете надежности?
8. Поясните влияние условий эксплуатации на надёжность электронного блока.

Исследование состояний и переходов для восстанавливаемой информационной системы

1. Цель работы

Изучение особенностей создания диаграмм состояний и переходов с использованием программ Simulink. Использование полученных навыков для исследования надежности восстанавливаемой системы.

2.1. Теоретическая часть

2.1. Марковский случайный процесс

При анализе надежности информационных систем широко используется понятие вероятностного (случайного) процесса.

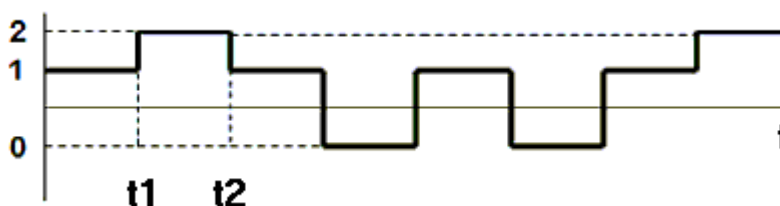
Существует большое число различных типов вероятностных случайных процессов. Наиболее подходящим для исследования надежности является марковский случайный процесс.

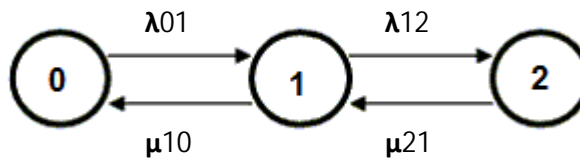
Марковский процесс – процесс, у которого для каждого момента времени вероятность любого состояния объекта в будущем зависит только от состояния объекта в настоящий момент времени и не зависит от того, каким образом объект пришел в это состояние.

Характеристику процесса полезно сопровождать графическим изображением. На рис. 1,а показан процесс изменения состояний некоторого объекта. Число возможных состояний – три. В момент времени t_1 объект переходит из состояния 1 в состояние 2, в момент времени t_2 – из состояния 2 в состояние 1 и т.д.

На рис. 1,б вероятностный процесс представлен графом состояний с указанием на ребрах графа интенсивностей переходов из одного состояния в другое.

В исследованиях надежности теория марковских процессов получила весьма широкое применение, т.к. процесс функционирования многих объектов, как правило, сопровождается простейшими потоками отказов и восстановлений. Экспоненциальное распределение времени работы до отказа и времени восстановления работоспособности – необходимое условие для марковского процесса.





б

Рис.1. Варианты изображения вероятностного процесса:

а – временной последовательностью смены состояний; б – графом состояний

В настоящее время возможно построение имитационных моделей (использующих диаграммы состояний и переходов) с использованием программ Stateflow и Simulink, входящих в состав пакета MATLAB.

Особое место среди инструментальных приложений занимает система визуального моделирования Simulink. Simulink хорош тем, что, с одной стороны, обеспечивает пользователю доступ ко всем основным возможностям пакета MATLAB, а с другой – является достаточно самостоятельной его компонентой, в том смысле, что при работе с ним не обязательно иметь навыки в использовании других инструментов, входящих в состав пакета.

Stateflow-инструмент для численного моделирования систем, характеризующихся сложным поведением.

Stateflow-диаграммы включаются в Simulink-модели, чтобы обеспечить возможность моделирования процессов, управляемых событиями. Stateflow обеспечивает ясное описание поведения сложных систем, используя диаграммы состояний и переходов.

Комбинация MATLAB-Simulink-Stateflow является мощным универсальным инструментом моделирования систем. Дополнительная возможность следить в режиме реального времени за процессом выполнения диаграммы путем включения режима анимации делает процесс моделирования реактивных систем по-настоящему наглядным.

2.2. Объекты Stateflow диаграммы

Stateflow диаграмма (Stateflow Chart) состоит из набора графических (состояния, переходы, соединения) и неграфических (события, данные, программные коды) объектов.

Состояние (State) – режим, в котором моделируемая система пребывает некоторое время, в течение которого она ведет себя одинаковым образом. Например, система может находиться в одном из двух состояний: работоспособном и неработоспособном. В диаграмме состояний и переходов состояния изображаются прямоугольниками с округленными углами.

Переход (Transition) – изменение состояния, обычно вызываемое некоторым событием. Переходы показываются линиями со стрелками, указывающими направление перехода. В большинстве случаев переход указывает на изменение состояния системы.

Событие (Event) – нечто происходящее в рассматриваемой системе или вне ее. Например, события могут быть связаны с фактами обнаружения отказа или восстановления элементов системы.

События управляют выполнением диаграммы Stateflow. Наступление события может запустить переход, и тогда он происходит, или может запустить действие, и тогда оно выполняется.

Действие (Action) – это результат выполнения какой-либо части диаграммы Stateflow.

Действие может быть выполнено в результате перехода от одного состояния к другому. Действие может быть также реакцией на состояние.

Внутри прямоугольника, обозначающего состояние, указываются метки. Метки состояний начинаются с имени состояния. Кроме того, метка состояния может иметь символ / (слэш) и одно или несколько ключевых слов. При наличии символа / (слэш) ключевое слово Entry может отсутствовать. Ключевые слова определяют различные типы действий, связанных с состоянием:

Entry – действия при входе в состояние;

During – действия во время активности состояния;

Exit - действия при выходе из состояния;

On event_name – действия при наступлении события event_name.

После ввода ключевого слова, которое определяет тип действия, необходимо указать, какие именно действия будут выполняться. Составные действия для каждого типа разделяются символом точка с запятой. После каждого ключевого слова ставится двоеточие.

На рис. 2 для состояния с именем Н0 указано действие $S=S+1$, которое выполняется при входе в это состояние.

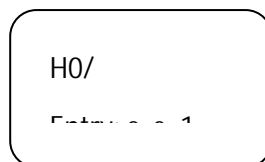


Рис. 2. Пример изображения состояния

Линии соединения могут разделяться на сегменты. Для этого используют объекты – соединения (узлы).

Соединение (Connective Junction) – графический объект, предназначенный для указания точки принятия решения в системе.

В следующем примере имеются два сегментированных перехода: один из состояния Н0 к состоянию Н1, а другой – от состояния Н0 обратно к состоянию Н0 (рис. 3).

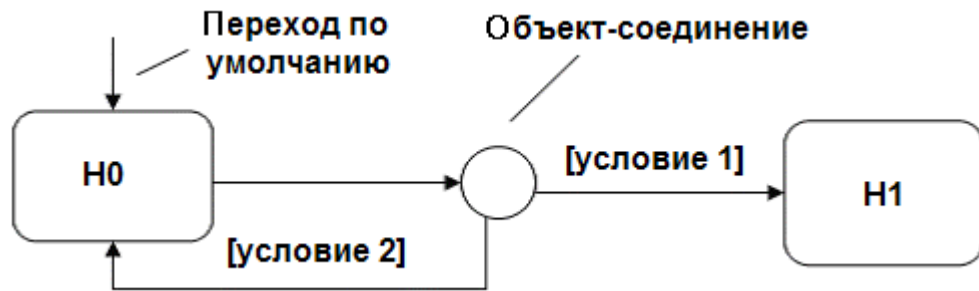


Рис. 3. Использование перехода (узла)

Переходы характеризуются метками. Метка может включать в себя имя события, условие, действие и/или действие перехода. Метки перехода имеют следующий формат:

Событие [условие] { действие условия}/ действие перехода.

Любая часть метки может отсутствовать.

Переход происходит при наступлении события, но с учетом истинности условия, если оно определено. Условия заключаются в квадратные скобки, а действия условий – в фигурные скобки. Действия условий выполняются перед тем как переход осуществится. Действия перехода выполняются после того, как переход стал возможен и при истинности условия. Действия перехода обозначаются символом (/).

Переход по умолчанию (рис. 3) определяет, какое из нескольких состояний должно быть активным, когда имеется неопределенность между двумя или более исключительными (ИЛИ) состояниями.

Когда состояние активно, диаграмма переходит в этот режим. Когда состояние неактивно, диаграмма находится не в этом режиме. Активность и неактивность состояний диаграммы динамически изменяется, базируясь на событиях и условиях.

В процессе моделирования используются различные данные. Например, значения интенсивностей переходов из одного состояния в другое должны поступать в Stateflow диаграмму от модели Simulink, а результаты расчетов вероятностей состояний марковского процесса должны выходить из Stateflow диаграммы в модель Simulink. *Объекты – данные* являются неграфическими объектами.

2.3. Создание объектов Stateflow диаграммы

Для открытия окна моделирования Stateflow диаграммы необходимо выполнить следующие действия:

- откройте окно MATLAB и щелкните мышью по кнопке Simulink;
- в браузере Simulink Library Browser выберите пункт Stateflow;
- выполните двойной щелчок левой кнопкой мыши по блоку Chart. Появляется окно Stateflow (chart) sflib/Chart, в котором создается Stateflow диаграмма.

Выделите на панели инструментов значок State (состояние), а затем щелкните мышью на свободном месте поля диаграммы. Повторяя эти действия, разместите на экране все блоки состояний.

Соедините блоки состояний линиями (стрелками) переходов. Для этого щелкните левой кнопкой мыши на линии, образующей блок, из которого должна выйти линия перехода, и не отпуская ее, переведите указатель мыши на линию, образующую блок, в который должна войти линия перехода.

Для каждого блока состояния модели введите соответствующие метки. Например, щелкните правой кнопкой мыши по первому блоку состояния модели и выберите пункт Properties. В поле Label окна State введите метку:

H0/
 $to0=ml(' -\log(\text{unifrnd}(0,1))')/ Io.$

В этой метке переменной to0 присваивается значение случайного числа, имеющего экспоненциальный закон распределения (логарифмическое преобразование применяется к случайным числам, подчиняющимся закону равномерной плотности). Переменная to0 моделирует время нахождения системы в работоспособном состоянии. Io – интенсивность отказов моделируемой системы (входной параметр модели Stateflow).

Для второго блока состояния модели введите метку:

H1/
 $tv1=ml(' -\log(\text{unifrnd}(0,1))')/ Iv.$

Переменная tv1 моделирует время нахождения системы в состоянии восстановления; Iv – интенсивность восстановления системы (входной параметр модели Stateflow).

Введите метки переходов. Например, щелкните правой кнопкой мыши по стрелке перехода из состояния H0 в состояние H1 и выберите пункт Properties.

В поле Label окна Transition введите метку

$\{Ts0=Ts0+to0; Tss=Tss+to0\}.$

Здесь в переменной Ts0 накапливается время нахождения системы в состоянии H0, а в переменной Tss – суммарное время моделирования.

Для стрелки перехода из состояния H1 в состояние H0 введите метку

$\{ Ts1=Ts1+tv1; Tss=Tss+tv1\}.$

В переменной Ts1 накапливается время нахождения системы в состоянии H1.

Создайте переход по умолчанию к состоянию H0. Для этого выделите на панели инструментов значок Default transition , а затем наведите указатель мыши на образующую линию блока H0.

В результате выполнения этих действий должна быть получена модель, изображенная на рис. 4.

После создания визуальных объектов модели необходимо определить объекты данных. Для этого щелкните правой кнопкой мыши по свободному месту диаграммы и выберите пункт Explore. Stateflow открывает окно Exploring. Выберите объект в иерархии объектов Проводника (например, Chart). Из меню Проводника Add выберите пункт Data. По умолчанию Stateflow добавляет определение для нового объекта в словарь данных (рис. 5).

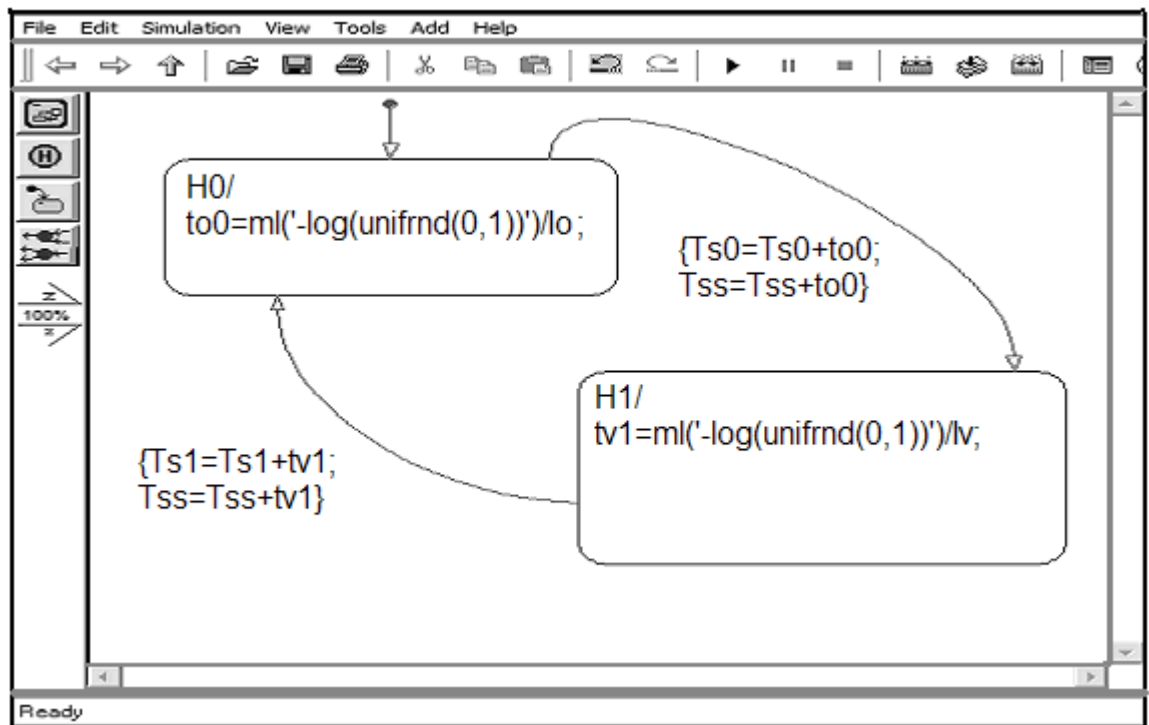


Рис. 4. Модель Stateflow


В столбце Name введите имя переменной, в столбце Scope – разновидность переменной (Local, Input, Output и т.п.). К разновидности Local относятся переменные, которые используются только внутри модели Stateflow, в данном примере к ним относятся `to0`, `tv1`. Переменные типа Input импортируются из Simulink (`Io`, `Iv`), а переменные типа Output являются выходными в Simulink модель, в примере это `Tss`, `Ts0`, `Ts1`.

Object Hierarchy		Contents of: (chart) lab2/Chart							
		Name	Scope	Trigger	Type	Size	Min	Max	InitVal
-	sflib								
	Chart								
+	untitled								
-	lab2								
	Chart	<code>Io</code>	Input(1)		double				
	h1	<code>Iv</code>	Input(2)		double				
	h0	<code>to0</code>	Local(1)		double				0
		<code>Ts0</code>	Output(1)		double				0
		<code>tv1</code>	Local(2)		double				0
		<code>Ts1</code>	Output(2)		double				0
		<code>Tss</code>	Output(3)		double				0

events(0) data(7) targets(0) | 7 [1:7]

Рис. 5. Окно Exploring

2.4. Создание объектов Simulink

Для установления связи созданной модели Stateflow с Simulink необходимо в главном окне, которое всегда активно после запуска Matlab, щёлкнуть левой кнопкой мыши по значку  в результате этого откроется окно библиотеки Simulink Library Browser (рис. 6).

В меню File выберите пункт new/model, после этого откроется окно untitled.

Чтобы связать ранее созданную модель Stateflow с моделью Simulink необходимо перенести блок Chart в область окна untitled модели Simulink (рис. 7). После этого в модель Stateflow можно входить, дважды щёлкнув левой кнопкой мыши по блоку Chart, помещённому в области Simulink .

На следующем этапе необходимо задать значение входных переменных Io и Iv. Для этого в окне Simulink Library Browser в разделе Simulink выберите блок Constant (смотри пункт Sources) и перенесите его в область модели Simulink. Соедините выход блока Constant с входным портом переменной Io. Щёлкните дважды левой кнопкой мыши по блоку Constant, в результате появится окно Block Parameters (рис. 8), в строке Constant value необходимо ввести нужное значение интенсивности отказов. Аналогичным образом создайте объект Constant1 и введите значение интенсивности восстановления.

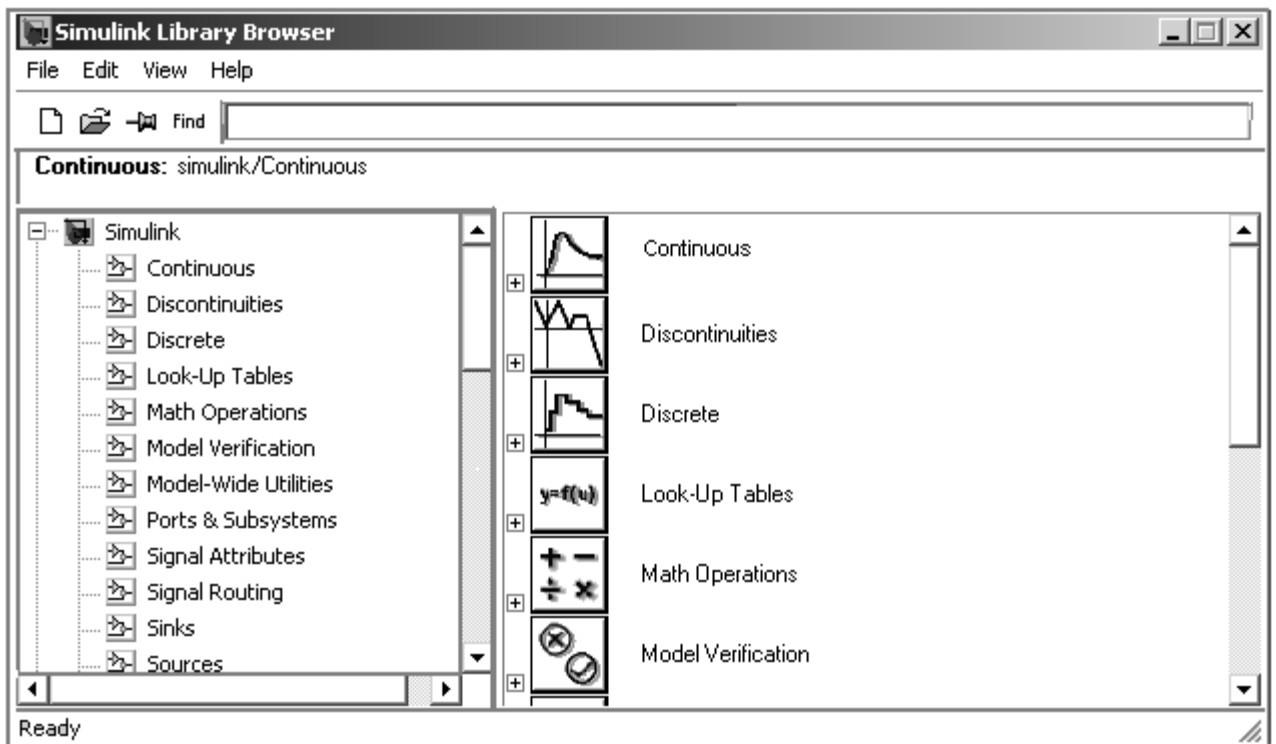


Рис. 6. Окно библиотеки Simulink Library Browser

Для отображения текущих значений переменных Ts0, Ts1, Tss используются три блока типа Display из раздела Sinks библиотеки Simulink. Входы этих блоков необходимо соединить с соответствующими выходными портами переменных блока Chart.

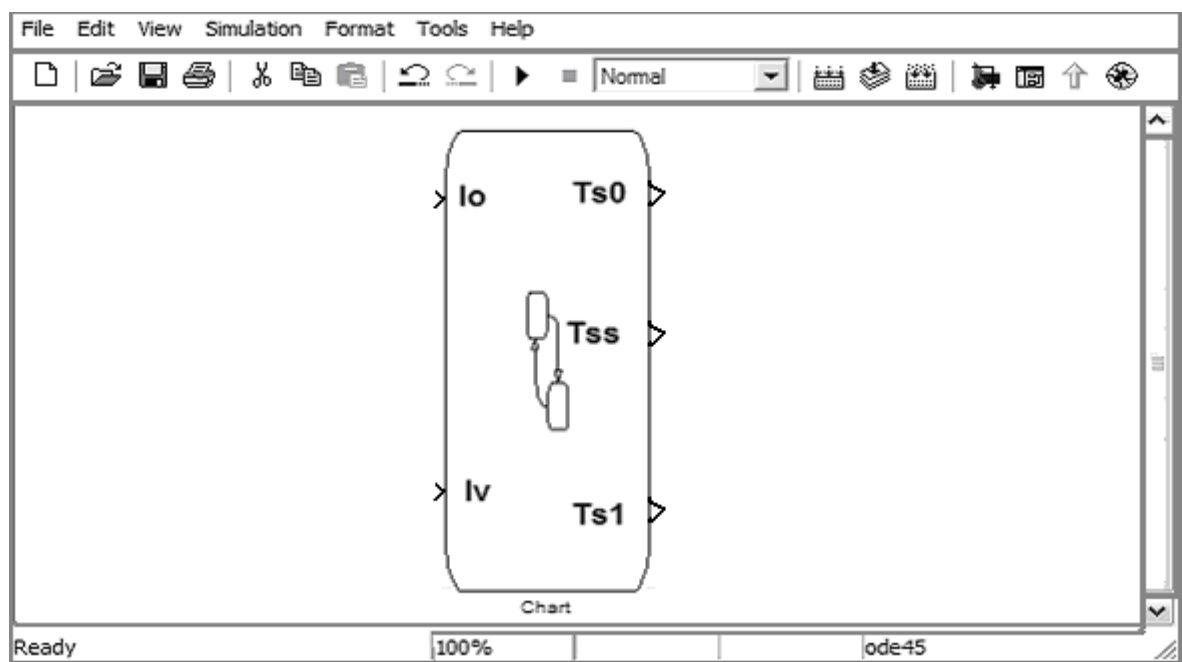


Рис. 7. Окно модели Simulink

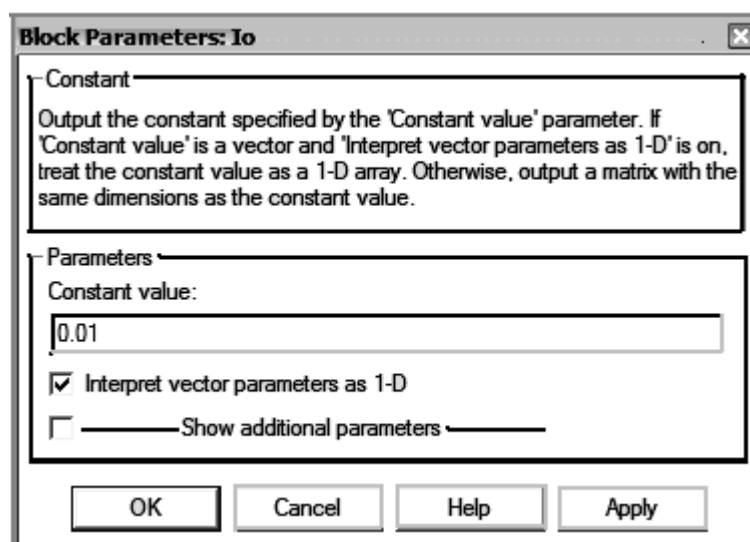


Рис. 8. Окно Block Parametrs блока Constant

Вероятности нахождения моделируемой системы в состояниях H_0 и H_1 определяются следующим образом:

$$P_{H_0} = T_{s0} / T_{ss}; \quad P_{H_1} = T_{s1} / T_{ss}.$$

Расчеты по этим формулам выполним с помощью двух блоков типа Product. Этот блок можно найти в разделе Math Operation. Чтобы задать операцию деления, щёлкните дважды по значку блока и в появившемся окне Block Parametrs в поле Number of inputs введите */ (рис. 9). После этого у блока появляются входы делимого и делителя. Соедините входы каждого из двух блоков типа Product с соответствующими выходами блока Chart.

Для отображения графиков вероятностей P_{H_0} и P_{H_1} воспользуемся блоком Scope (см. раздел Simulink пункт sinks). Этот блок моделирует многолучевой осциллограф. Щёлкните дважды по этому блоку и в меню выберите пункт Parametrs, открывается окно, изображенное на рис.10, в поле Number of axes задайте значение 2. Соедините входы блока Scope с выходами блока Product. По оси ординат шкалы измерений блока Scope откладываются значения наблюдаемой величины, по оси абсцисс – значения модельного времени. По умолчанию для оси ординат используется диапазон [-5;5], а для оси модельного времени – [0;10]. Для изменения масштаба можно использовать кнопки Zoom X-axis, Zoom Y-axis или кнопку Autoscale, которая позволяет автоматически установить оптимальный масштаб осей (автошкалирование).

Вид результирующей модели показан на рис. 11.

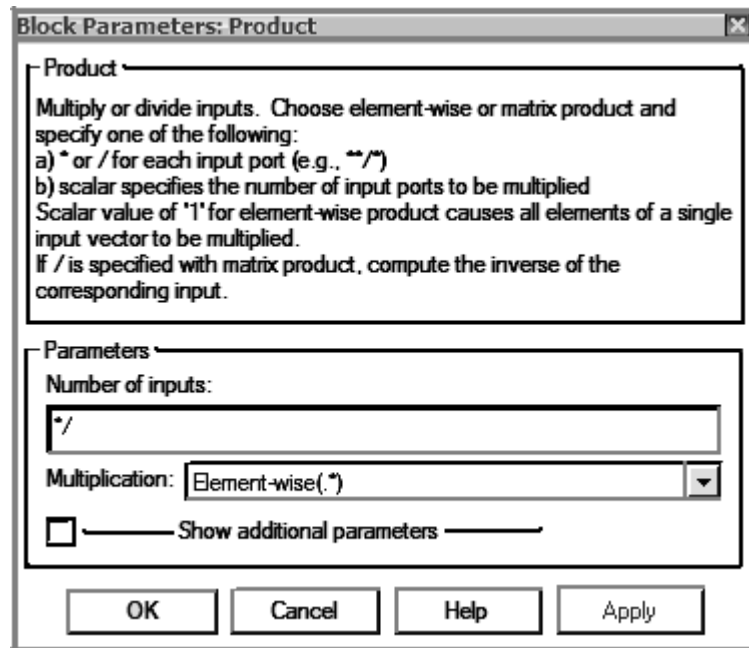


Рис. 9. Окно Block Parameters блока Product

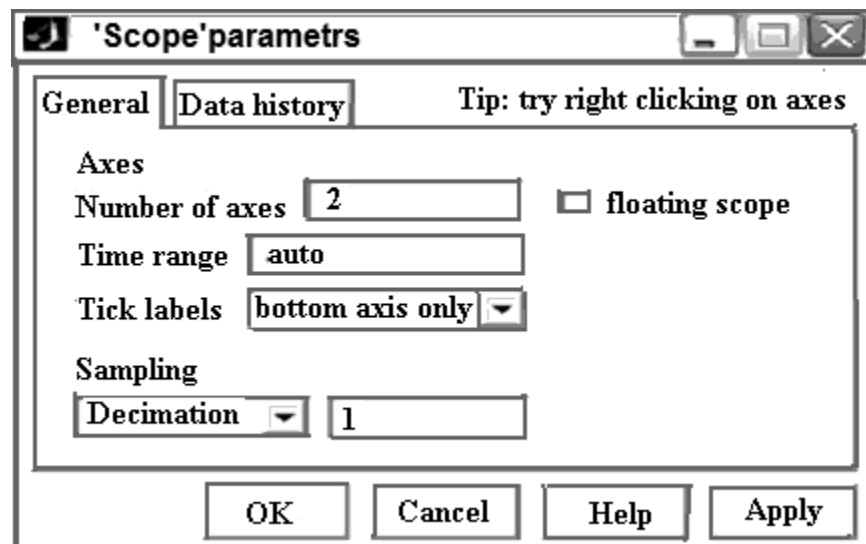



Рис. 10. Окно Parameters блока scope

После добавления всех необходимых блоков и соединения их между собой надо задать время симуляции. Выберите в меню пункт Simulation / Simulation parameters, в результате появится окно Simulation parameters (рис.12). И в полях Start time и Stop time задайте необходимый интервал времени симуляции, а в поле Fixed step size – величину шага моделирования.

Запустите модель на выполнение, нажав значок  на панели окна Stateflow. Процесс моделирования сопровождается мерцанием блоков состояний и линий переходов. После окончания моделирования необходимо открыть модель Simulink для просмотра и анализа результатов. В блоках типа Display отображаются окончательные значения для выходных переменных: Ts0, Ts1, Tss.

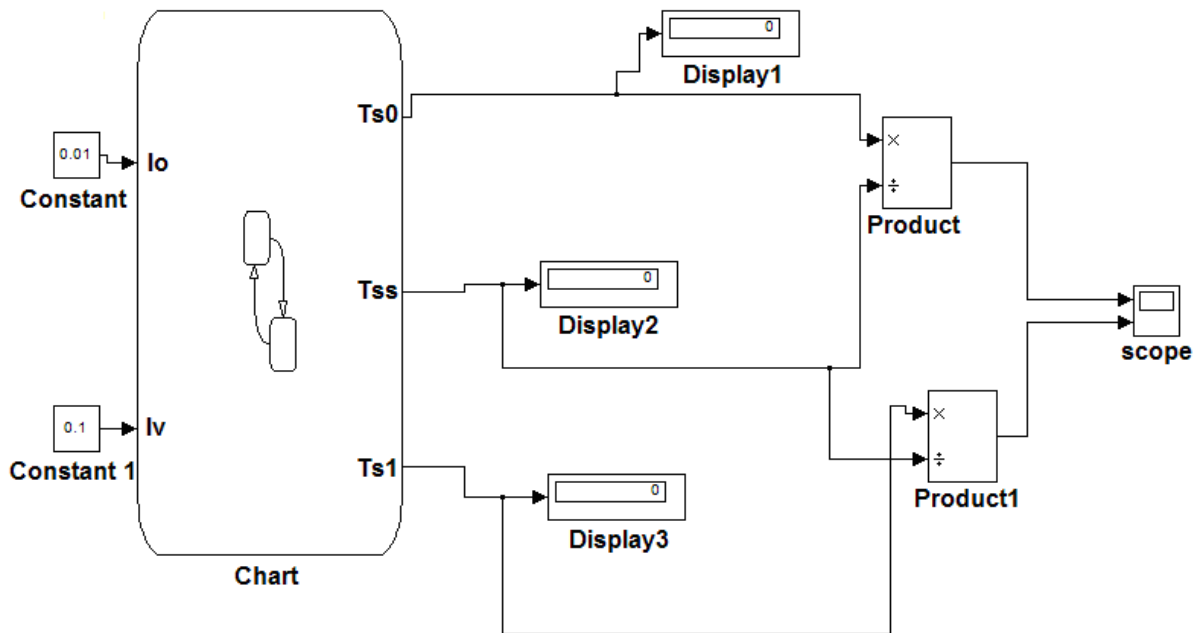


Рис. 11. Модель Simulink

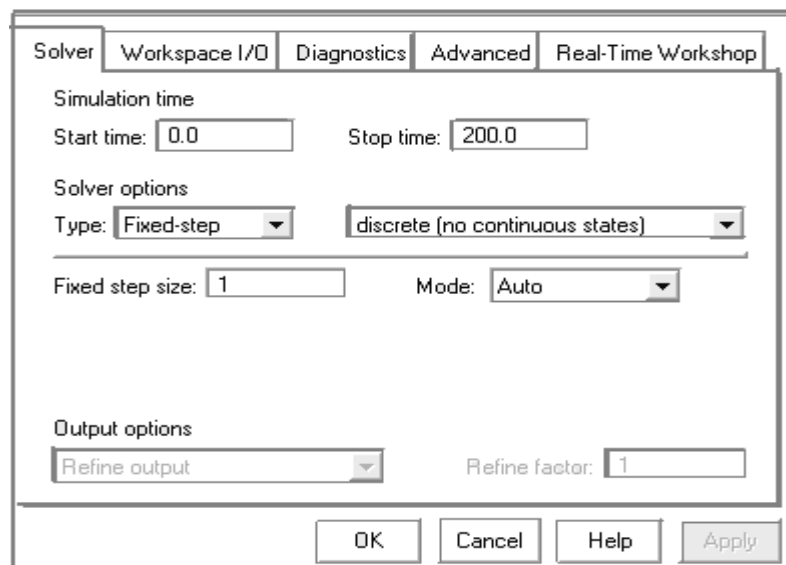


Рис. 12. Окно Simulation parameters

Щёлкните дважды мышью по блоку Scope, появится окно, в котором должны отображаться два графика: $P_{H0}(n)$ и $P_{H1}(n)$, где n – номер шага моделирования. Для отображения этих графиков щёлкните мышью по кнопке Autoscale.

3. Порядок выполнения лабораторной работы

1. Создать модели Simulink и Stateflow(рис. 4, рис. 11).
2. Задать значения интенсивностей отказов и восстановлений моделируемой системы. При этом необходимо воспользоваться следующими рекомендациями:

$$I_o=0.001+B/1000$$
, где B – номер бригады;

$$I_v=I_o*K$$
, где K принимает значение из ряда (1, 2, 4).
3. Задать параметры моделирования (рис. 12).
4. Снять графики изменения вероятностей $P_{H0}(n)$ и $P_{H1}(n)$ при различных значениях I_v .
5. Определить значения вероятностей P_{H0} и P_{H1} в установившемся режиме.
6. Определить теоретические значения коэффициента готовности и коэффициента вынужденного простоя системы.
7. Сравнить результаты, полученные в пп. 5, 6.
8. Измените модели, изображённые на рис. 4 и рис. 11, так, чтобы появилась возможность расчёта значений средней наработки на отказ и среднего времени восстановления.
9. Составить отчет по лабораторной работе. Отчёт должен содержать модели Simulink и Stateflow, исходные данные для моделирования, результаты моделирования, оформленные в виде таблиц и графиков.

4. Контрольные вопросы

1. Дайте определение марковского процесса.
2. Каковы основные объекты Stateflow диаграммы?
3. Какой формат имеют метки?
4. Как создать Stateflow диаграмму?
5. Каким образом задаются объекты данных?
6. Как связать модель Stateflow с моделью Simulink?
7. Как задать значения входных переменных модели Simulink?
8. Напишите формулы, которые позволяют найти коэффициенты готовности и коэффициенты вынужденного простоя восстанавливаемой системы.
9. Поясните действия, необходимые для задания интервала времени симуляции и шага моделирования.
10. Расскажите, какие основные блоки библиотеки Simulink используются для создания модели?

Лабораторная работа №3

Исследование резервированной восстанавливаемой информационной системы

1. Цель работы

Изучение особенностей моделирования и анализа надежности дублированной системы с использованием программ Stateflow и Simulink.

2. Теоретическая часть

2.1. Резервированная восстанавливаемая система как система массового обслуживания

Пусть кратность резервирования системы m , а заданная наработка t . В процессе работы подсистемы могут отказывать, и система изменяет свои состояния (H_0, H_1, H_2, \dots). Отказавшие подсистемы восстанавливаются и включаются в работу, т.е. при восстановлении работоспособного состояния отказавшей подсистемы резервированная подсистема переходит из состояния H_j (отказало j подсистем) в состояние H_{j-1} (отказало $j-1$ подсистем). Если потоки отказов и восстановлений простейшие, то процесс, протекающий в резервированной системе, можно представить как процесс системы массового обслуживания в виде марковской схемы (рис. 1).

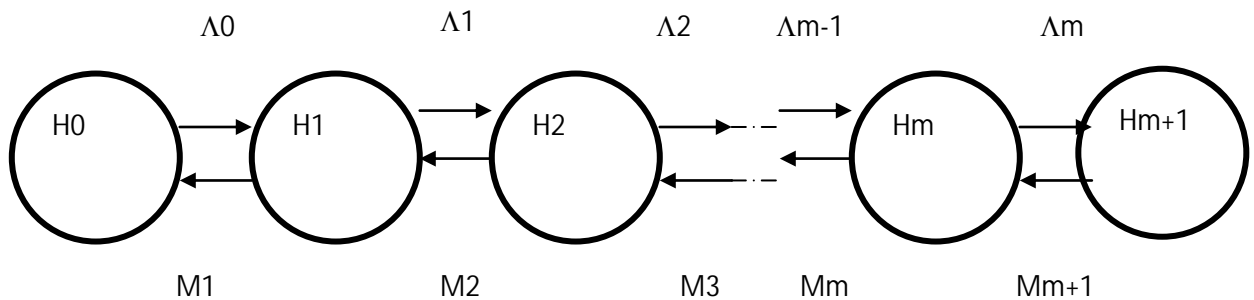


Рис. 1. Марковская схема

Стрелки – линии перехода показывают возможные пути перехода системы из состояния H_j в состояние H_{j+1} при отказе одной из подсистем или – в состояние H_{j-1} при восстановлении работоспособного состояния отказавшей подсистемы. Λ_j и M_j соответственно характеризуют интенсивность такого перехода.

Если интенсивность перехода $M_{m+1}=0$, то система, попав в состояние H_{m+1} , не может изменить своего состояния в дальнейшем и называется *системой с поглощающим экраном*.

По характеру восстановления подсистем различают резервированные системы с полностью ограниченным, частично ограниченным и неограниченным восстановлением.

При полностью ограниченном восстановлении интенсивность переходов $M_j = \mu$ ($j = \overline{1, m+1}$); при частично ограниченном восстановлении

$$M_j = \begin{cases} j\mu & \text{при } j < k, \\ k\mu & \text{при } j \geq k; \end{cases} \quad (1)$$

при неограниченном восстановлении $M_j = j\mu$, где μ - интенсивность восстановления одной подсистемы; k - коэффициент, величина которого соответствует максимальному числу отказавших подсистем, которые могут параллельно восстанавливаться.

В лабораторной работе исследуется дублированная восстанавливаемая система с отражающим экраном, т.е. $m=1$ и $M_2 \neq 0$.

В процессе исследования используются два способа:

- моделирование с помощью программ Stateflow и Simulink;
- решение системы дифференциальных уравнений Колмогорова.

2.2. Особенности создания Stateflow диаграмм

Stateflow диаграмма должна содержать три блока (рис. 2):

H_0 – обе подсистемы дублированной восстанавливаемой системы находятся в работоспособном состоянии;

H_1 – одна из двух подсистем находится в неработоспособном состоянии;

H_2 – обе подсистемы находятся в неработоспособном состоянии.

Переменная to_0 моделирует время нахождения дублированной системы в состоянии H_0 , т.е. после окончания этого отрезка времени происходит отказ одной из подсистем.

Переход из состояния H_0 в состояние H_1 выполняется безусловно. При каждом переходе осуществляется накопление значений двух переменных:

Ts_0 – суммарное время пребывания системы в состоянии H_0 ;

Tss – суммарное время моделирования.

После входа системы в состояние H_1 генерируются значения переменных:

tv_1 – время восстановления работоспособного состояния отказавшей подсистемы;

to_1 – время отказа подсистемы, оставшейся в работоспособном состоянии после отказа первой подсистемы.

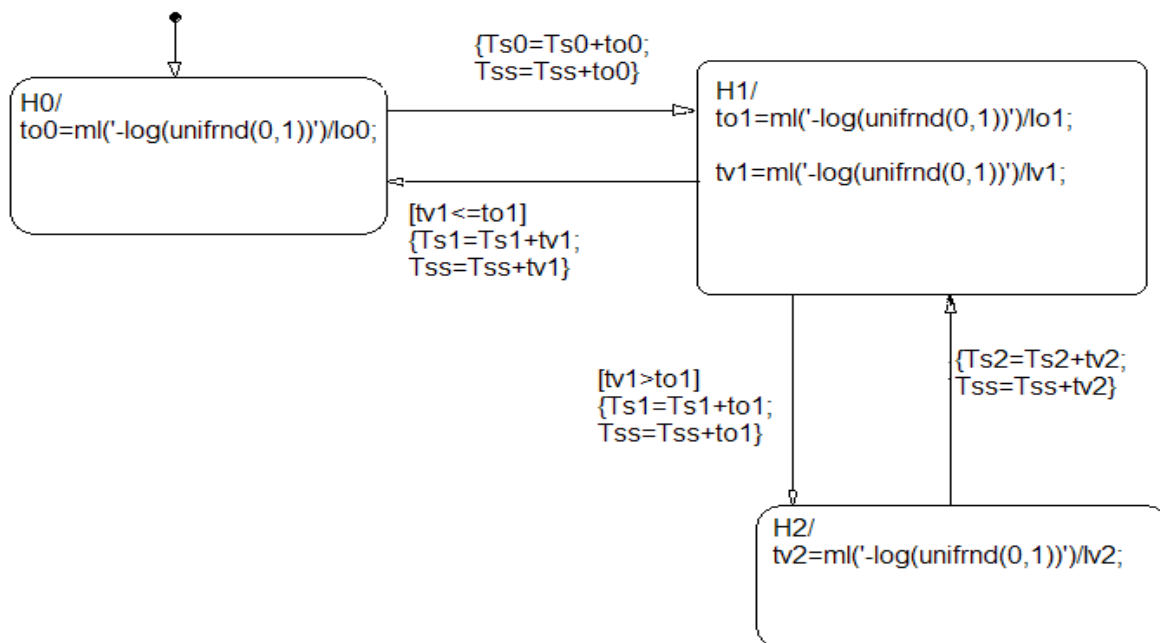


Рис. 2. Stateflow диаграмма модели

Если $tv1 \leq to1$, то происходит переход в состояние H0, при этом происходит накопление значения переменных Ts1 (суммарное время пребывания системы в состоянии H1) и переменной Tss.

Если $tv1 > to1$, то система переходит в состояние H2. Это означает, что одна из отказавших подсистем не была восстановлена до момента отказа другой подсистемы. В этом случае происходит модификация значений переменных Ts1 и Tss.

Переменная tv2 моделирует время восстановления одной из двух отказавших подсистем. Из состояния H2 система безусловно переходит в состояние H1, при этом происходит накопление значения переменных Ts2 (суммарное время пребывания системы в состоянии H2) и Tss.

Переменные Io0 и Io1 в модели Stateflow соответствуют интенсивностям отказов Λ_0, Λ_1 в марковской схеме, а переменные Io0 и Io1 – интенсивностям восстановления M1 и M2. Значения этих переменных должны быть заданы в модели Simulink.

Выходными переменными являются Ts0, Ts1, Ts2 и Tss.

Вероятности нахождения моделируемой системы в состояниях H0, H1 и H2

$$P_{H0} = Ts0/Tss; \quad P_{H1} = Ts1/Tss; \quad P_{H2} = Ts2/Tss. \quad (2)$$

Коэффициент готовности

$$Kг = P_{H0} + P_{H1} = (Ts0 + Ts1) / Tss. \quad (3)$$

2.3. Особенности создания объектов в Simulink

На рис. 3 изображена схема одного из возможных вариантов задания значений интенсивностей отказов и восстановлений.

С помощью блока Constant задается значение интенсивности отказов λ одной подсистемы. Блок Product1 выполняет функцию умножающего устройства. Умножение на коэффициент 2 (блок Constant1) необходимо для моделирования дублированной системы с нагруженным резервом.

На выходе переключателя Manual Switch (он находится в разделе Signal Routing) формируется необходимое значение интенсивности отказов λ_0 .

Совокупность блоков Constant1, Constant2, Product2 и переключателя Manual Switch1 позволяет получить соответствующие значения интенсивностей восстановления. Блок умножения Product2 используется для моделирования дублированной восстанавливаемой системы с неограниченным восстановлением. Нажатие правой кнопкой мыши по блоку Manual Switch позволяет изменять позиции переключателя для задания различных вариантов интенсивностей отказов и интенсивностей восстановлений. Для вычисления вероятностей P_{H0} , P_{H1} и P_{H2} необходимо ввести в модель Simulink еще три блока типа Product, а для отображения их значений – три блока Display. Отображение соответствующих графиков можно выполнить с помощью трехканального блока Scope.

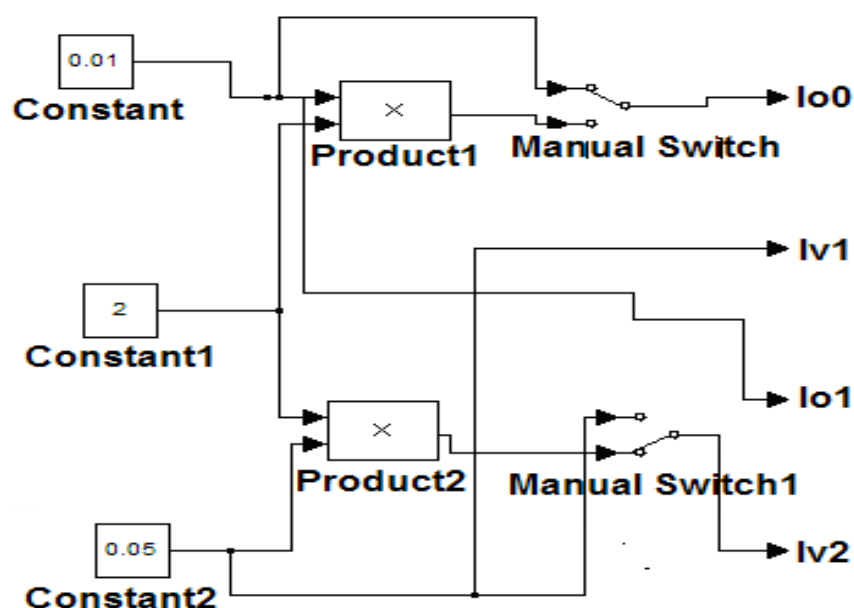


Рис. 3. Схема варианта задания

2.4. Составление системы дифференциальных уравнений Колмогорова

Каждому из состояний H_0 , H_1 и H_2 указывается вероятность ($P_0(t)$, $P_1(t)$ и $P_2(t)$) того, что система в момент времени t будет находиться в данном состоянии.

Методика расчета показателей надежности сводится к следующему:

- 1) составляется система дифференциальных уравнений в соответствии с графом состояний системы;
- 2) выбираются начальные условия решения задачи;
- 3) определяются показатели надежности системы.

Система уравнений составляется согласно правилу: *производная вероятности данного состояния равна алгебраической сумме произведений интенсивностей всех возможных переходов этого состояния на вероятность состояний, из которых выходят линии перехода. Знак у слагаемого положительный, если линия перехода входит в данное состояние, и отрицательный, если линия перехода выходит из этого состояния.*

Согласно этому правилу составим дифференциальные уравнения для дублированной системы:

$$\left. \begin{aligned} \frac{dP_0(t)}{dt} &= -\Lambda_0 P_0(t) + M_1 P_1(t), \\ \frac{dP_1(t)}{dt} &= \Lambda_0 P_0(t) - (M_1 + \lambda_1) P_1(t) + M_2 P_2(t), \\ \frac{dP_2(t)}{dt} &= \Lambda_1 P_1(t) - M_2 P_2(t). \end{aligned} \right\} \quad (4)$$

Система дифференциальных уравнений (4) называется системой уравнений Колмогорова или системой уравнений массового обслуживания.

Состояния системы H_0 , H_1 и H_2 в момент времени t образуют полную группу несовместных состояний. Тогда сумма вероятностей этих состояний

$$P_0(t) + P_1(t) + P_2(t) = 1. \quad (5)$$

Начальные условия задаются следующими: при $t=0$ вероятность $P_0(0)=1$, $P_1(0)=0$ и $P_2(0)=0$.

Если система дублирована, то ее коэффициент готовности

$$K_r = \left(1 + \frac{\Lambda_0}{M_1}\right) \left(1 + \frac{\Lambda_0}{M_1} + \frac{\Lambda_0 \Lambda_1}{M_1 M_2}\right)^{-1}. \quad (6)$$

Для дублированной системы с нагруженным резервом

$$\Lambda_0 = 2\lambda; \Lambda_1 = \lambda, \quad (7)$$

а при ненагруженном

$$\Lambda_0 = \Lambda_1 = \lambda. \quad (8)$$

Для дублированной системы с неограниченным восстановлением

$$M_1 = \mu; M_2 = 2\mu, \quad (9)$$

а при ограниченном

$$M1 = M2 = \mu. \quad (10)$$

2.5. Решение системы дифференциальных уравнений в системе MATLAB

Решение обыкновенных дифференциальных уравнений рассмотрим на примере системы уравнений, с заданными параметрами: $\Lambda_0=2, \Lambda_1=1, M_1=1, M_2=2$, тогда система уравнений (4) примет вид:


$$\left. \begin{aligned} \frac{dP_0(t)}{dt} &= -2P_0(t) + P_1(t), \\ \frac{dP_1(t)}{dt} &= 2P_0(t) - (1+1)P_1(t) + 2P_2(t), \\ \frac{dP_2(t)}{dt} &= P_1(t) - 2P_2(t). \end{aligned} \right\} \quad (11)$$

Для решения обыкновенных дифференциальных уравнений (ODU) в MATLAB могут быть применены численные методы. MATLAB может выполнять последовательность операторов, записанных в файл на диске. Такие файлы называются М-файлами, потому что имена файлов имеют вид <имя>.m. Большая часть работы в MATLAB состоит в создании, редактировании и выполнении таких m-файлов. Существует два типа m-файлов: файлы – сценарии (М-сценарии) и файлы – функции (М-функции).

Общий порядок программирования решения системы ODU:

- 1) создается М-функция с описанием правых частей дифференциальных уравнений;
- 2) создается М-сценарий с выбранным решателем.

Для создания М-функции необходимо в меню системы нажать на значок

 в результате откроется окно редактирования (рис. 4), в котором можно набирать необходимый текст М-функции или сценария.

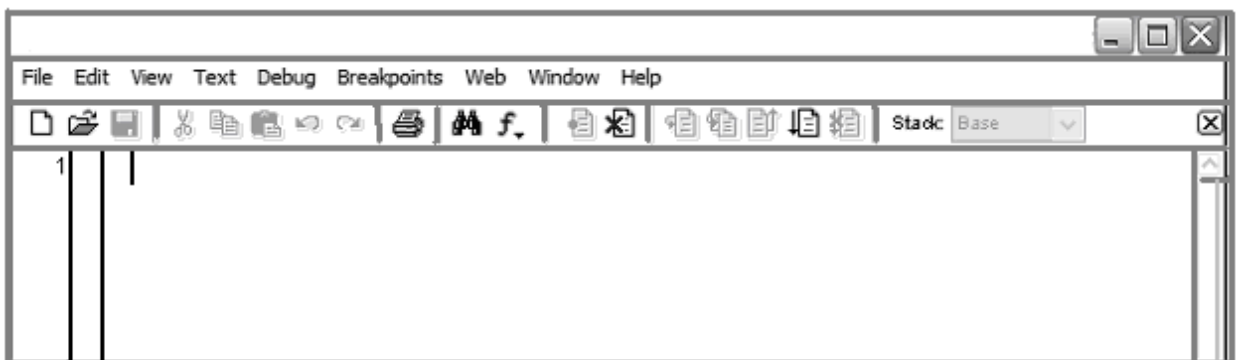


Рис. 4. Окно редактирования

Создадим М-функцию под именем `odu.m` (имя может быть выбрано любым) для системы уравнений (11). Обозначим функцию переменной `du`. В системе это будет выглядеть так:

```
function du=odu(t,p);  
du=[-2*p(1)+ p(2);  
     2*p(1)-(1+1)*p(2)+2*p(3);  
     1*p(2)-2*p(3)].
```

В данной записи отображается правая часть системы (11). Строки выражения разделяются точкой с запятой. Символ `P` выступает в роли переменной. В круглых скобках записывается ее порядковый номер. Например, значению `P(1)` соответствует $P_0(t)$, `P(2)` - $P_1(t)$, и т.д.

После создания М-файла, его необходимо сохранить под тем же именем, что и название функции, в данном случае, `odu.m`.

В `MATLAB` для решения обыкновенных дифференциальных уравнений предусмотрены специальные функции-решатели, например `ode23`, `ode45`.

Синтаксис:

```
[t, P] = ode23('<имя функции>', T, pn)  
[t, P] = ode45('<имя функции>', T, pn).
```

Входные параметры:

'<имя функции>' - строковая переменная, являющаяся именем М-файла, в котором вычисляются правые части системы дифференциальных уравнений (15);

T – интервал интегрирования системы дифференциальных уравнений;

pn - вектор начальных условий для вероятностей состояний $P_0(t)$, $P_1(t)$, $P_2(t)$;

Выходные параметры:

t - текущее время;

P - многомерный массив, где каждый столбец соответствует одной переменной.

М-сценарий создаётся в отдельном файле, повторив те же действия как для создания М-функции. Для нашего примера текст сценария будет следующим:

```
T=[0 5];  
pn=[1;0;0];  
[t,p]=ode45('odu',T,pn);  
plot(t,p),grid,title('lab'),legend('p0','p1','p2').
```

В первой строке данного выражения задаётся интервал интегрирования T, который зависит от значений интенсивностей отказов и восстановлений. Конкретнее значения определяются методом последовательного приближения. Зададим интервал от 0 до 5

(интервал задаётся через пробел). Во второй строке определяются начальные условия p_0 , которые зададим последовательностью чисел 1,0,0. Третья строка вызывает решатель ode45. С помощью функции plot задаются оси координат для построения графика функций вероятностей $P_0(t)$, $P_1(t)$, $P_2(t)$. Функция grid строит на графике сетку. Функция title выводит название графика. Legend служит для задания названий графикам функций.

Далее необходимо сохранить M-сценарий, например, под именем `odu1`, и выйти в командную строку MATLAB.

Чтобы запустить на выполнение созданный M-сценарий нужно в активной командной строке набрать имя M-сценария (в рассматриваемом примере – `odu1`) и нажать клавишу Enter. В результате на экране появится график функций (рис. 5).

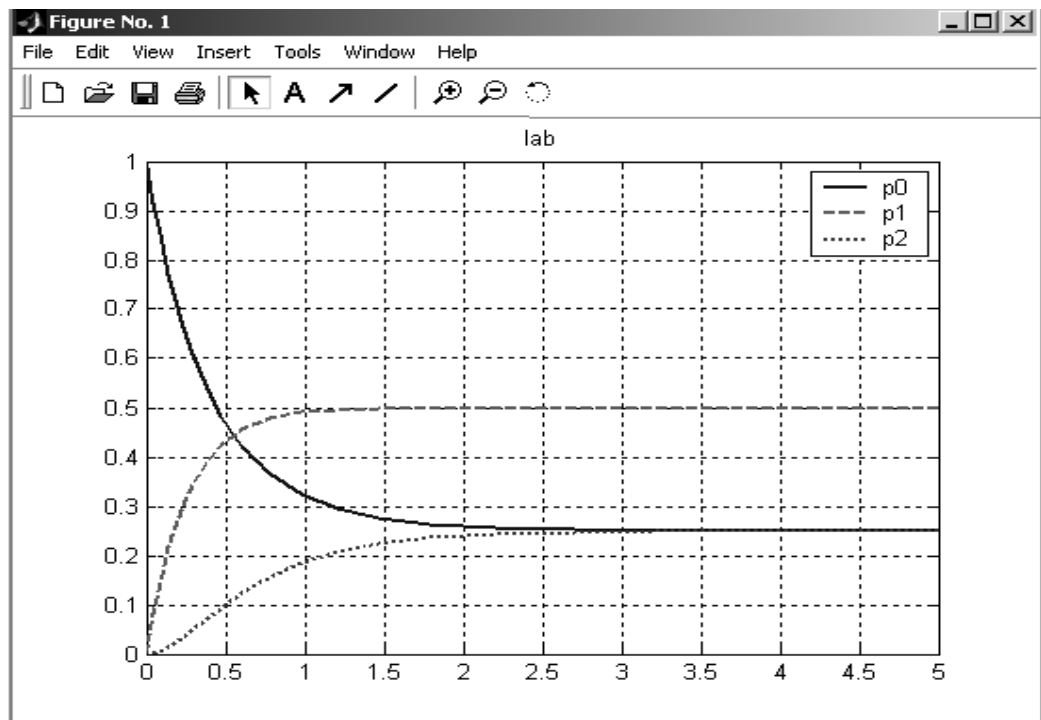


Рис. 5. График полученного решения

Массив с полученными решениями можно посмотреть, выбрав в меню VIEW/WORKSPACE, и в левой части окна щёлкнуть на значок массива P (рис. 6.), в отдельном окне откроется массив размером 30x3 со значениями результатов решения (рис. 7.). Также в данном окне можно найти начальные условия p_0 , интервал интегрирования T и значения времени t.

Name	Size	Bytes	Class
T	1x2	16	double array
p	30x3	720	double array
pn	3x1	24	double array
t	30x1	240	double array

Рис. 6. Массивы данных

	1	2	3
1	1	0	0
2	0.99992	7.9994e-005	1.5999e-009
3	0.99952	0.00047977	5.7572e-008
4	0.99752	0.0024739	1.5338e-006
5	0.98764	0.012326	3.8454e-005
6	0.96294	0.036711	0.00034983
7	0.93868	0.060346	0.00096974
8	0.91488	0.083229	0.0018927

Рис. 7. Массив результатов решения

3. Порядок выполнения лабораторной работы

3.1. Исследование объектов Stateflow и Simulink

1. Создать модели Stateflow и Simulink.
2. Задать значения интенсивностей отказов и восстановления (соответствующие рекомендации приведены в описании к лабораторной работе №2), при этом необходимо учитывать соотношения (7)-(10).
3. С помощью переключателей Product1, Product2 задать один из четырех типов моделируемой системы [см. (7)-(10)].
4. Снять графики $P_{H0}(n)$, $P_{H1}(n)$ и $P_{H2}(n)$.
5. Рассчитать теоретическое значение коэффициента готовности (6) и значение (3), полученное в результате моделирования. Сравнить полученные результаты.
6. Исследовать четыре типа моделируемой системы. Сравнить их по надежности.

3.2. Решение дифференциальных уравнений в системе Matlab

1. Составить системы дифференциальных уравнений для четырех типов моделируемой системы.
2. Создать M–сценарий и M–функцию для решения уравнений.
3. Получить графики функций $P_{H0}(n)$, $P_{H1}(n)$ и $P_{H2}(n)$.
4. Сравнить полученные результаты с результатами п. 3.1.
5. Составить отчет по лабораторной работе. Отчет должен содержать модели Simulink и Stateflow, исходные данные и результаты моделирования, системы дифференциальных уравнений Колмогорова и результаты их решения.

4. Контрольные вопросы

1. Поясните процесс функционирования дублированной восстанавливаемой системы.
2. Дайте определения системе с поглощающим и отражающим экраном.
3. Как найти коэффициент готовности?
4. Как определяются интенсивности отказов Λ_0 и Λ_1 для систем с нагруженным и ненагруженным резервом?
5. Как определяются интенсивности восстановления для систем с ограниченным и неограниченным восстановлением?
6. Поясните условия переходов моделируемой системы из состояния H1.
7. Как при моделировании определяются вероятности нахождения системы в различных состояниях?
8. Как с помощью переключателей (рис. 3) задать нужный вариант моделируемой системы?
9. Какой из четырех исследуемых вариантов является наиболее надежным?
10. Сформулируйте правило составления систем дифференциальных уравнений Колмогорова.
11. Как решить систему дифференциальных уравнений в системе Matlab?

Лабораторная работа №4
**Исследование эффективности функционирования дублированной
восстанавливаемой информационной системы**

1. Цель работы

Изучение возможности применения программ Simulink для исследования эффективности функционирования дублированной восстанавливаемой системы.

2. Теоретическая часть

2.1. Понятие эффективности функционирования

Рассматривая надежность технической системы, обычно интересуются ее способностью выполнять заданные функции. При этом предполагается, что система может находиться только в работоспособном состоянии или только в неработоспособном. Качественная сторона функционирования системы при выполнении какой-либо задачи не рассматривается.

Однако многие современные системы (сбора, передачи, хранения, отображения и обработки информации) имеют структуру, которая обеспечивает при возникновении отказов отдельных элементов, звеньев или подсистем частичное работоспособное состояние системы. Такие системы после возникновения отдельных отказов продолжают функционировать, выполняя заданные функции не в полном объеме или с ухудшением качества функционирования.

Эффективность функционирования такой системы характеризует степень приспособленности ее к выполнению заданных функций при определенных условиях, степень целесообразности ее применения.

В качестве примера рассмотрим систему сбора и передачи информации, у которой имеются два независимых канала связи. Она может находиться в одном из четырех состояний (рис. 1):

Н0 – состояние, при котором оба канала связи работоспособны;

Н1 и Н2 – состояния, при которых соответственно неработоспособен или канал 1, или канал 2;

Н3 – состояние, при котором и канал 1, и канал 2 неработоспособны.

В процессе работы система из-за отказов каналов связи (и их восстановления) переходит из одного состояния в другое, причем каждому из состояний свойственен вполне определенный выходной эффект. Каждое такое состояние системы количественно может характеризоваться условным показателем эффективности функционирования.

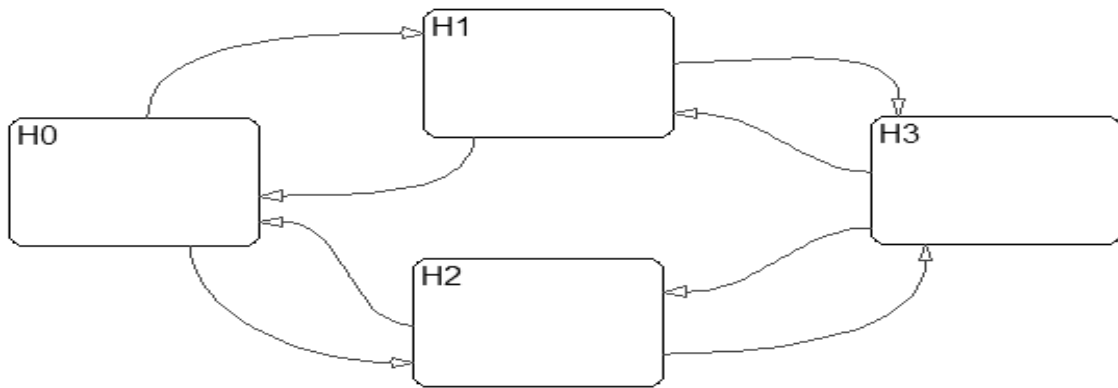


Рис. 1. Состояния системы

Под условным показателем эффективности функционирования системы, находящейся в определенном состоянии, понимают значение выходного эффекта, создаваемого этой системой, при условии, что она в заданный момент времени находится именно в этом состоянии.

Очевидно, наибольшая эффективность функционирования системы (рис. 1) будет, если работоспособны оба канала связи (H0), и выходной эффект отсутствует при двух отказавших каналах (H3).

Пусть E_j – условный показатель эффективности функционирования системы, при условии, что она находится в состоянии H_j ($j=0,3$). Например, $E_0=10^6$ бит/с, $E_1=5 \cdot 10^5$ бит/с, $E_2=5 \cdot 10^5$ бит/с, $E_3=0$.

Обозначим T_j – суммарное время нахождения системы в состоянии H_j , рассчитанное для заданного момента времени t .

Тогда выражение для оценки показателя эффективности функционирования системы в момент времени t

$$E_c(t) = E_0 \cdot T_0 + E_1 \cdot T_1 + E_2 \cdot T_2 + E_3 \cdot T_3 . \quad (1)$$

Если разделить левую и правую части этого выражения на $E_0 \cdot \sum_{j=0}^3 T_j$, то получим относительный показатель эффективности функционирования, называемый коэффициентом эффективности функционирования системы, т.е.

$$E_{c0}(t) = \frac{\sum_{j=0}^3 E_j T_j}{E_0 \sum_{j=0}^3 T_j} . \quad (2)$$

2.2. Особенности построения Stateflow диаграммы и модели Simulink

В процессе моделирования должна быть создана Simulink диаграмма, соответствующая рис.1. При моделировании используются знания, полученные при выполнении лабораторных работ №2 и №3.

Для моделирования случайного времени отказа канала целесообразно использовать переменные вида t_{oks} , где k – номер отказавшего канала ($k=1,2$); s – номер состояния системы, в котором произошел отказ канала ($s=0,2$). Например, случайное число, соответствующее времени отказа первого канала в состоянии H_0 , определяется переменной

$$t_{o10} = \text{ml}(-\log(\text{unifrnd}(0,1))) / I_{o1},$$

где I_{o1} – интенсивность отказа первого канала.

Для моделирования времени восстановления работоспособного состояния канала рекомендуется использовать переменные вида t_{vks} , где k – номер восстанавливаемого канала ($k=1,2$); s – номер состояния системы, в котором произошло восстановление отказавшего канала ($s=1,3$). Например, для моделирования времени восстановления второго канала в третьем состоянии необходимо использовать переменную

$$t_{v23} = \text{ml}(-\log(\text{unifrnd}(0,1))) / I_{v2},$$

где I_{v2} – интенсивность восстановления второго канала.

Для каждого перехода необходимо ввести соответствующую метку, в которой указываются условие перехода и действие, которое должно быть выполнено при переходе. Действия связаны с определением суммарного времени T_j нахождения системы в состоянии H_j ($j=0,3$) и общего времени моделирования. Например, метка перехода системы из состояния H_0 в состояние H_1 имеет следующий вид:

$$[t_{o10} \leq t_{o20}] \{T_{s0} = T_{s0} + t_{o10}; T_{ss} = T_{ss} + t_{o10}\},$$

где T_{s0} – переменная, соответствующая суммарному времени нахождения системы в состоянии H_0 ; T_{ss} – переменная, в которой накапливается общее время моделирования.

После создания Stateflow диаграммы необходимо разработать модель Simulink (рис. 2).

С помощью блоков Constant – Constant3 вводятся в Simulink модель (блок Chart) значения входных переменных, т.е. интенсивности отказов (I_{o1} , I_{o2}) и интенсивности восстановлений (I_{v1} , I_{v2}) каналов связи. Выходными переменными блока Chart являются T_{s0} – T_{s3} . В блоках Product – Product3 значения этих переменных умножаются на условные показатели E_j эффективности функционирования системы, формируемые блоками Constant4 – Constant7.

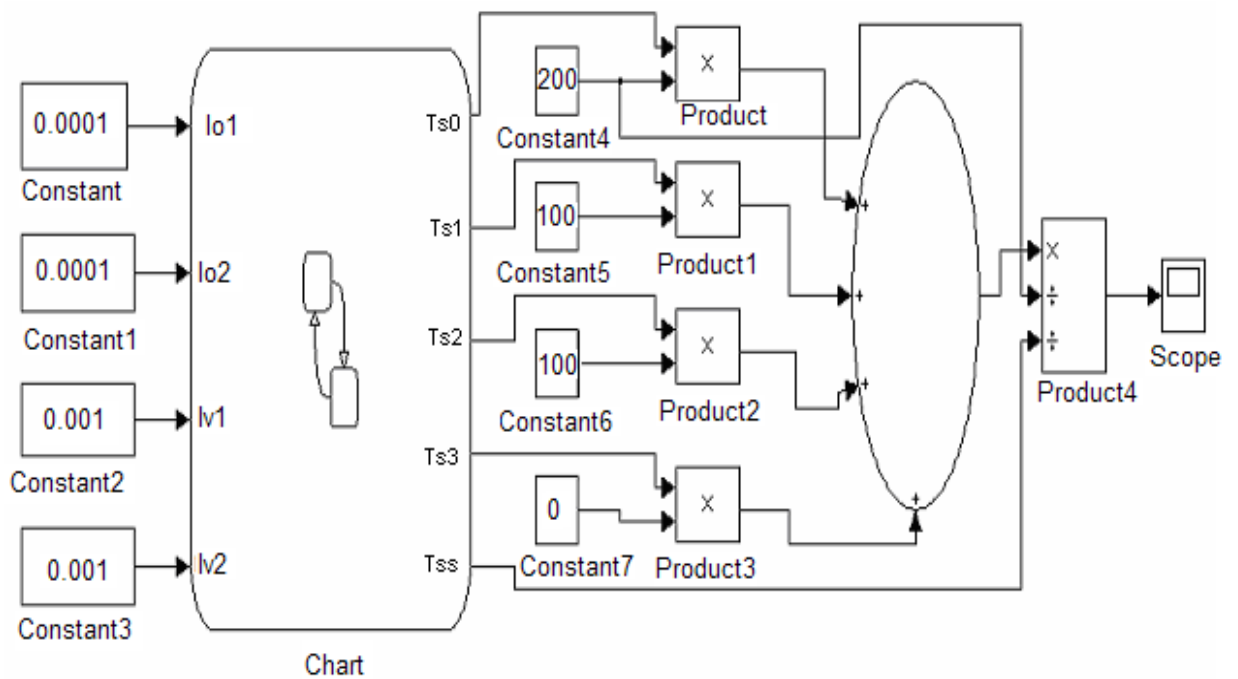


Рис. 2. Модель Simulink

На выходе сумматора определяется значение числителя формулы, а с помощью блока Product4 выполняется деление полученного результата на E0 и Tss, т.е. вычисляется относительный показатель эффективности функционирования. Результаты моделирования отображаются на экране блока Scope.

3. Порядок выполнения лабораторной работы

1. Выбрать предметную область, в которой предполагается использовать дублированную восстанавливаемую систему.
2. Создать модели Stateflow и Simulink, считая, что моделируемая система относится к классу систем с отражающим экраном.
3. Задать значения интенсивностей отказов и интенсивностей восстановлений подсистем:

$$Io1=2Io2, \text{ или } Io1=Io2, \text{ или } Io2=2Io1;$$

$$Iv1=2Iv2, \text{ или } Iv1=Iv2, \text{ или } Iv2=2Iv1.$$

Рекомендации по выбору Io и Iv приведены в описании к лабораторной работе №2.

4. Задать значения условных показателей эффективности функционирования системы:

$$E1=1000 \cdot B, \text{ где } B - \text{ номер бригады};$$

$$E2 = 2E1 \text{ или } E2=0,5E1;$$

$$E0 = E1+E2.$$

5. Задать параметры моделирования.
6. Снять график зависимости коэффициента эффективности от числа шагов моделирования $Es0(n)$.

7. Исследовать влияние параметров I_0, I_v, E_1, E_2, E_0 на величину коэффициента эффективности.
8. Составить отчет по лабораторной работе. Отчет должен содержать описание выбранной предметной области, диаграмму состояния системы, модели Simulink и Stateflow, результаты моделирования, оформленные в виде таблицы и графиков.

4. Контрольные вопросы

1. Обоснуйте выбранную предметную область.
2. Поясните понятие эффективности функционирования выбранной предметной области.
3. Дайте определение условного показателя эффективности функционирования.
4. Напишите выражение для оценки показателя эффективности функционирования.
5. Как рассчитать относительный показатель эффективности функционирования?
6. Поясните условия переходов между состояниями для схемы, изображенной на рис. 1.
7. Как рассчитывается время нахождения моделируемой системы в различных состояниях?
8. Поясните принцип действия разработанной модели Simulink.
9. Укажите способы увеличения коэффициента эффективности функционирования.
10. Поясните график зависимости коэффициента эффективности функционирования от числа шагов моделирования.
11. При каких условиях величина коэффициента эффективности функционирования приближается к единице?

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

	Авторы,	Заглавие	Издательство, год	Адрес	
Л1.1	Блинков, Ю. В.	Основы теории информационных процессов и систем: учебное пособие	Пенза: Пензенский государственный университет архитектуры и строительства, ЭБС АСВ, 2011	http://www.iprbookshop.ru/23103.htm	
Л1.2	Белов, П. С.	Математическое моделирование технологических процессов: учебное пособие (конспект лекций)	Егорьевск: Егорьевский технологический институт (филиал) Московского государственного технологического университета «СТАНКИН», 2016	http://www.iprbookshop.ru/43395.htm	
6.1.2. Дополнительная литература					
	Авторы,	Заглавие	Издательство, год	Адрес	

Л2.1	Казиев В. М.	Введение в анализ, синтез и моделирование систем	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ),	http://www.iprbookshop.ru/52188.htm	
Л2.2	Плохотников, К. Э.	Методы разработки математических	Москва: СОЛОН-ПРЕСС, 2017	http://www.iprbookshop.ru/64926.htm	
6.1.3. Методические разработки					
	Авторы,	Заглавие	Издательство, год	Адрес	
Л3.1	Татарникова, Т. М.	Моделирование систем: методические указания к выполнению	Санкт-Петербург: Российский государственный гидрометеорологический университет, 2008	http://www.iprbookshop.ru/12503.htm	
Л3.2	Шевцова, Ю. В.	Математические модели и методы исследования операций: сборник задач	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2009	http://www.iprbookshop.ru/54766.htm	
Л3.3	Сёмина, В. В.	Моделирование систем: методические указания для проведения	Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2016	http://www.iprbookshop.ru/64869.htm	
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"					
Э1	Душин В.К. Теоретические основы информационных процессов и систем [Электронный ресурс]: учебник/ Душин В.К.— Электрон. текстовые данные.— М.: Дашков и К, 2014.— 348 с.— ББК 37.01. ЭБС АСВ. URL: http://www.iprbookshop.ru/24764				
Э2	Шатрова Г.В. Методы исследования и моделирования информационных процессов и технологий [Электронный ресурс]: учебное пособие/ Шатрова Г.В., Топчиев И.Н.— Электрон. текстовые данные.— Ставрополь: Северо-Кавказский федеральный университет, 2016.— 180 с.— ББК 37.01. ЭБС АСВ. URL: http://www.iprbookshop.ru/64869.htm				
Э3	Лубенец Ю.В. Экономико-математические методы и модели [Электронный ресурс]: учебное пособие/ Лубенец Ю.В.— Электрон. текстовые данные.— Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013.— 64 с.— ББК 37.01. ЭБС АСВ. URL: http://www.iprbookshop.ru/64869.htm				



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Методы исследования и моделирования
информационных процессов и технологий»
для студентов направления подготовки
09.04.02 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Методы исследования и моделирования информационных процессов и технологий» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1.

Лабораторная работа 2.

Лабораторная работа 3.

Лабораторная работа 4.

Лабораторная работа 5.

Лабораторная работа 6.

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины - развить системное мышление у обучающихся путем детального анализа подходов к математическому моделированию и сравнительного анализа разных типов моделей. Ознакомить обучающихся с математическими свойствами методов и моделей оптимизации, которые могут использоваться при анализе и решении широкого спектра задач. Выработать у обучающихся навыки проведения численных исследований математических моделей и анализа результатов вычислений. Научить выбирать наиболее перспективное управляющее решение.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

ОПК-8.1: Анализирует современные методологии разработки программных средств и проектов, требования, стандарты и принципы составления технической документации, методы управления коллективом разработчиков

ОПК-5.3: Модернизирует программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач

ОПК-1.3: Использует приемы теоретического и экспериментального исследования объектов профессиональной деятельности, в том числе в новой или незнакомой среде и в междисциплинарном контексте

Изучив данный курс, студент должен:

Знать:

- методы системного анализа и программной инженерии;
- способы сбора и анализа научно-технической информации.

Уметь:

- выполнять анализ и исследовать информационные процессы;
- разрабатывать методику анализа информационных процессов и технологий;
- выполнять моделирование информационных процессов и технологий

Владеть:

- экспериментально-теоретическими методами исследования;
- методами моделирование систем и процессов.

Реализация компетентного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа 1

Освоение графоаналитического метода исследования информационных процессов

Цель работы:

1. Закрепить .
2. Научиться составлять .

1 Задание на лабораторную работу

- 1.1 Создание графа
- 1.2 Построение матрицы смежности

2 Краткие теоретические сведения

2.1 Графоаналитический метод

Графоаналитический метод основан на анализе матрицы смежности информационного графа. В данном случае исходными для анализа информационных потоков являются данные о парных отношениях между наборами информационных элементов, формализуемые в виде матрицы смежности. Под **информационными элементами** понимают различные типы входных, промежуточных и выходных данных.

Например, графическая модель для задачи преобразования данных может быть представлена в виде двудольного графа, состоящего из множества узлов-операций, соединенных дугами с множеством узлов — информационных элементов, как показано на рисунке 1.

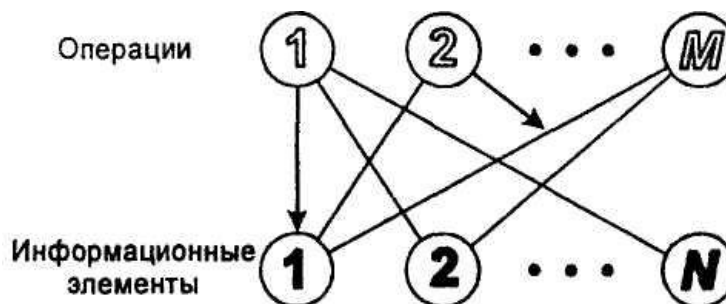


Рисунок 1 - Граф преобразования данных

Этот граф можно сделать *раскрашенным*, т.е пометить различным цветом дуги, относящиеся к разным информационным элементам. Тогда задача минимизации информационных потоков в виде графа будет состоять в разбиении раскрашенного графа на подграфы (модули), при котором минимизируется суммарное число дуг различного цвета, связывающих выделенные подграфы.

2.2 Матрица смежности

Матрица смежности — квадратная бинарная матрица с количеством строк (и столбцов), равным количеству информационных элементов. В каждой позиции матрицы смежности записывают единицу, если между соответствующими элементами матрицы существует отношение, то есть информация одного документа используется при формировании другого, и в соответствующей позиции ставится ноль, если отношения нет. Далее матрице смежности ставится в соответствие граф информационных взаимосвязей. Множеством вершин графа является множество информационных элементов, дуги отражают взаимосвязи между элементами. Дуга присутствует, если в матрице смежности отношение между элементами отмечено единицей, и отсутствует в противном случае. Анализ графа позволяет выявить информационную зависимость между входными, промежуточными и выходными документами, характер зависимости, установить направление движения информации. Графоаналитический метод является развитием метода с использованием сетевой модели и позволяет проводить более детальный анализ информационных потоков.

Например, для удобства математического описания задачи управления процедурой преобразования и метода ее решения сведем граф, показанный на рисунке 1, к табличной форме, расположив по строкам выполняемые операции, а по столбцам — элементы множества идентификаторов исходных, промежуточных и выходных данных, связанных с выполнением этих операций.

На пересечении строки и столбца ставится 1, если операция и информационный элемент связаны. Другими словами, получим матрицу L :

$$L = \begin{matrix} & \begin{matrix} D_1 & D_2 & \dots & D_n \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ \dots \\ A_m \end{matrix} & \begin{vmatrix} l_{11} & l_{12} & \dots & l_{1n} \\ l_{21} & l_{22} & \dots & l_{2n} \\ \dots & \dots & \dots & \dots \\ l_{m1} & l_{m2} & \dots & l_{mn} \end{vmatrix} \end{matrix},$$

где $l_{ij} = 1$ — если информационный элемент D_j используется при выполнении операции A_i ;
 $l_{ij} = 0$ — в противном случае;
 $i = 1, \dots, m; j = 1, \dots, n$.

При таком представлении задача состоит в разбиении множества операций преобразования данных матрицы L на непересекающиеся подмножества (модули), суммарное число информационных связей между которыми минимально. При решении задачи должны быть учтены ограничения:

- на число выделяемых подмножеств (модулей);
- на число информационных элементов, входящих в один модуль;
- на число информационных связей между выделяемыми модулями;
- на совместимость операций в модулях.

Данная задача может быть сведена к задаче линейного программирования и решена с использованием стандартных прикладных программ.

3 Методические указания по выполнению работы

3.1 Постановка задачи

Описание потоков информации процессов управления с помощью графов является одним из наиболее разработанных методов. С его помощью достигается наглядность

функционирования системы управления и движения потоков информации; применение математического аппарата теории графов позволяет оптимизировать работу управления и каналов связи; имеется возможность также представить динамику управления и движения информации, которая ускользает при использовании другими методами. В настоящее время имеется много примеров использования теории графов в описании данных процессов. Они различаются по характеру описываемых объектов, по видам графов. Рассмотрим наиболее типовые примеры.

Исследование информационного потока на основе его сетевой модели. Метод основан на применении сетевого графика и традиционных методов его анализа и оптимизации. Понятия работы и события принимают характер, соответствующий процессу управления, а именно: под работой понимается определенная задача управления, решаемая работниками данного аппарата управления; под событием понимается определенный документ, который был составлен в ходе выполнения работ (конечное событие) либо будет использоваться в ходе выполнения работ (начальное событие).

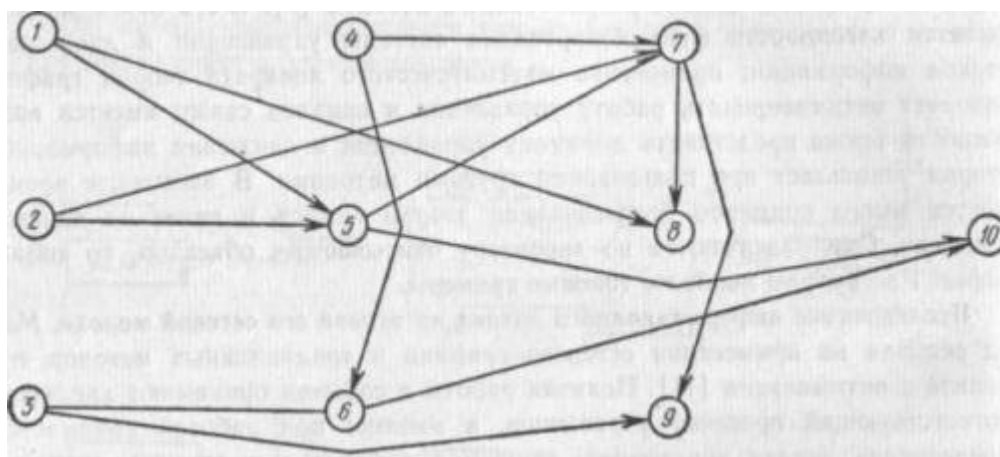
Анализ сетевой модели управления производится традиционными методами. Находятся критический путь, резервы времени (каждая работа, производимая управленческим аппаратом, характеризуется своей длительностью, определено время наступления каждого события), определяются узкие места, производится перераспределение ресурсов и т. д.).

3.2 Создание графа

Графоаналитический метод исследования потоков информации основан на построении информационного графа и анализе его матрицы смежности.

В любой управляющей системе различаются входы, выходы и внутренняя память. Через входы управляющая система получает исходные Данные из внешней среды, через входы во внешнюю среду выдаются результаты работы системы. В процессе функционирования управляющей системы появляется промежуточное звено между исходными данными и результатами Функционирования. Все три звена вместе образуют компоненты потока информации — x_i . Между компонентами потока информации существует упорядоченность. Так, нулевой порядок имеет исходные данные, наивысший — результаты функционирования.

На основании такой схемы потоков информации можно построить граф, вершинами которого служат x_i — компоненты потока информации и которые соединяются дугами в том случае, если переход между ними осуществляется без каких-либо промежуточных результатов (в противном случае недоопределена вершина). Дуги ориентируются в направлении результатов более высокого порядка. Построенный граф называется информационным. Общий вид этого графа показан на рисунке 1.



3.3 Построение матрицы смежности

Матрица смежности для графа строится следующим образом: элемент (i,j) , стоящий на пересечении i -й строки и j -го столбца, равен единице, если из вершины v_i в вершину v_j идет дуга, и равен нулю в противном случае. Матрица смежности является компактной моделью информационного графа. В дальнейшем строится последовательность матриц, представляющих собой матрицу смежности, возведенную в квадрат, третью степень и т. д. Общее количество матриц равно порядку информационного графа.

Матричная модель позволяет определить:

порядок схемы потока информации;

порядок каждой компоненты потока;

число компонент, непосредственно участвующих в формировании каждого результата;

число результатов, в формировании которых непосредственно участвует каждая компонента;

число путей фиксированной длины, связывающих любые две компоненты потока;

число возможных путей, связывающих любые две компоненты потока;

все результаты, для формирования которых используется каждая компонента, и все компоненты, необходимые для формирования каждого результата;

номер такта, после которого может быть погашена во внешней памяти каждая компонента исходных данных и промежуточных результатов;

число тактов, в течение которых каждая компонента хранится во внешней памяти.

Порядок графа и функциональных результатов является, таким образом, некоторой оценкой потока информации. Время погашения и продолжительности хранения может использоваться при анализе организации и объема внешней памяти.

Рассмотрим условный пример графа, представленного на рисунке 1.

Составим матрицу смежности и последовательность матрицы по степеням.

		M_1												M_2											
		1	2	3	4	5	6	7	8	9	10			1	2	3	4	5	6	7	8	9	10		
1					1				1			2	1							1	1		1	3	
2				1		1						2	2							1	2	1	1	5	
3					1					1		3	3									1		1	
4						1	1				1	4	4							1	1	1		3	
5							1	1				5	5							1	1			1	
6												6	6											0	
7										1	1	7	7											0	
8												8	8											0	
9												9	9											0	
10												10	10											0	
		0 0 0 0 2 2 3 3 2 2												0 0 0 0 0 0 2 5 3 4											

		M_3												Матрица числа путей												
		1	2	3	4	5	6	7	8	9	10			1	2	3	4	5	6	7	8	9	10			
1									1	1		2	1							1	1	3	1	1	3	
2									1	1		2	2							1	2	3	2	1	5	
3												3	0							1		1	1	1	1	
4												4	0							1	1	1	1	1	3	
5												5	0								1	2	1	1	1	
6												6	0											1	1	
7												7	0								1	1	1		1	
8												8	0												0	
9												9	0												0	
10												10	0												0	
		0 0 0 0 0 0 0 0 2 2 0																								

Анализ этих матриц позволяет определить данные по всем перечисленным выше пунктам. Например, проанализировав результирующую строку каждой матрицы по числу нулей, можно судить о порядке компонент. Для X_1, X_2, X_3, X_4 число входящих путей равно 0 (нули в таблице M_3), следовательно, эти компоненты являются исходными данными 0-порядка. В таблице M_2 образовались нули для компонент X_5, X_6 и в третьей таблице — для X_7 и X_{10} . Четвертая таблица степени M_3 не проводилась, так как она равна нулю по всем элементам. Итак,

- X_1, X_2, X_3, X_4 — компоненты 0-порядка;
- X_5, X_6 — компоненты 1-го порядка;
- X_7, X_{10} — компоненты 2-го порядка;
- X_8, X_9 — компоненты 3-го порядка.

Упорядочим граф по тактам движения потоков информации, как показано на рисунке 2. Таким образом, анализ матрицы смежности позволил выяснить подробности структуры графа, не замеченные при первом рассмотрении.

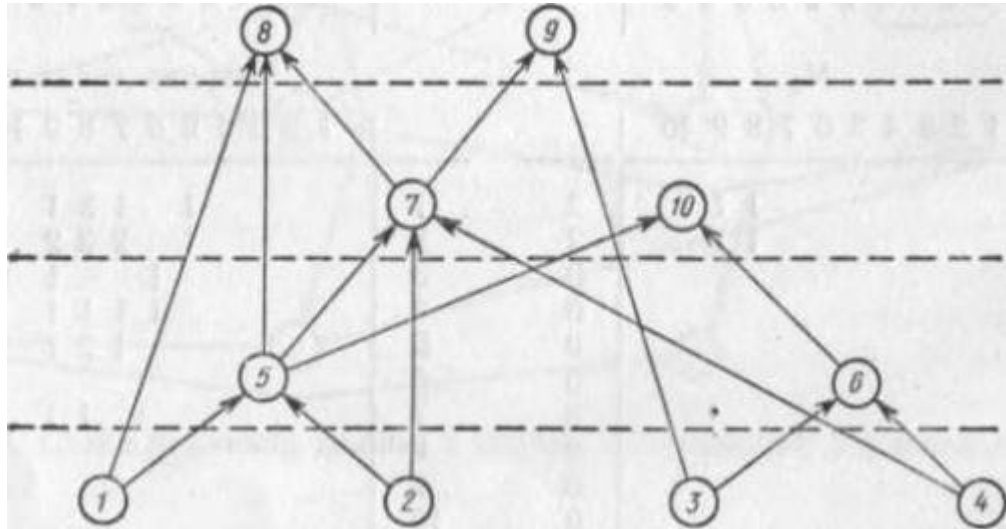


Рисунок 2 - Граф, упорядоченный по тактам движения потоков информации

Дополнительно.

Примечание

Контрольные вопросы

- 1 Поясните назначение графоаналитического метода.
- 2 Какие разновидности графов можно использовать?
- 3 Какова структура матрицы смежности?

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

6.1.1. Основная литература		
Заглавие	Издательство, год	Адрес
Методы исследования и моделирования информационных процессов и технологий: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2016	http://www.iprbookshop.ru/63100.html
Моделирование систем. Подходы и методы: учебное пособие	Санкт-Петербург: Санкт-Петербургский политехнический университет Петра Великого, 2013	http://www.iprbookshop.ru/43957.html
6.1.2. Дополнительная литература		
Заглавие	Издательство, год	Адрес
Математическое моделирование систем: учебное пособие	Томск: Томский государственный университет систем управления и радиоэлектроники, 2014	http://www.iprbookshop.ru/72124.html
Методы компьютерного моделирования физических процессов и сложных систем: учебное пособие	Санкт-Петербург: Университет ИТМО, 2014	http://www.iprbookshop.ru/67284.html
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"		
Э1	Методические рекомендации	
Э2	Селетков С.Н. Аналитические исследования в информационных и коммуникационных средах [Электронный ресурс]: учебное пособие/ Селетков С.Н., Днепровская Н.В.— Электрон. текстовые данные.— М.: Евразийский открытый институт, 2010.— 184 с.— Режим доступа:	
Э3	Блинков Ю.В. Основы теории информационных процессов и систем [Электронный ресурс]: учебное пособие/ Блинков Ю.В.— Электрон. текстовые данные.— Пенза: Пензенский государственный университет архитектуры и строительства, ЭБС АСВ, 2011.— 184 с.— Режим доступа: http://www.iprbookshop.ru/23103.html .— ЭБС «IPRbooks»	



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Средства автоматизированного проектирования информационных
систем»

для студентов направления подготовки

09.04.02 Информационные системы и технологии

Направленность (профиль) Информационные системы и технологии

Методические указания по дисциплине «Средства автоматизированного проектирования информационных систем» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1.

Лабораторная работа 2.

Лабораторная работа 3.

Лабораторная работа 4.

Лабораторная работа 5.

Лабораторная работа 6.

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины - развить системное мышление у обучающихся путем детального анализа подходов к математическому моделированию и сравнительного анализа разных типов моделей. Ознакомить обучающихся с математическими свойствами методов и моделей оптимизации, которые могут использоваться при анализе и решении широкого спектра задач. Выработать у обучающихся навыки проведения численных исследований математических моделей и анализа результатов вычислений. Научить выбирать наиболее перспективное управляющее решение.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

ОПК-2.1: Использует современные информационнокоммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач

ОПК-5.1: Анализирует современное программное и аппаратное обеспечение информационных и автоматизированных систем

ОПК-7.3: Создает математически модели для реализации успешного функционирования распределенных информационных систем и систем поддержки принятия решений

Изучив данный курс, студент должен:

Знать:

- структуру средств автоматизированного проектирования ИС,
- критериев качества средств автоматизированного проектирования ИС,
- современные средства автоматизированного моделирования информационных систем,
- методику внедрения САПР ИС.

Уметь:

- использовать средства автоматизированного моделирования информационных систем,
- внедрять САПР ИС в производство, -разрабатывать новые средства проектирования ИС.

Владеть:

- методикой применения САПР ИС при их разработке,
- методикой использования САПР ИС для управления проектом.

Реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа 1

Анализ возможностей интегрированного комплекса CASE-средств AllFusion Modeling Suite

Цель работы:

1. Закрепить основные способы использования CASE-средства.
2. Научиться анализировать функциональные возможности CASE-средства.

2 Краткие теоретические сведения

2.1 Состав CASE-средства AllFusion Modeling Suite

AllFusion Modeling Suite (ранее: **ERwin Modeling Suite**) - интегрированный комплекс CASE-средств, обеспечивающий все потребности компаний-разработчиков ПО. Данный пакет служит для проектирования и анализа баз данных, бизнес-процессов и информационных систем и включает продукты: [AllFusion Process Modeler](#) (ранее: **BPwin**), [AllFusion ERwin Data Modeler](#) (ранее: **ERwin**), [AllFusion Data Model Validator](#) (**ERwin Examiner**), [AllFusion Model Manager](#) (ранее: **ModelMart**), [AllFusion Component Modeler](#) (**Paradigm Plus**), использование которых позволяет сократить расходы и повысить продуктивность процесса разработки.

AllFusion Process Modeler

AllFusion Process Modeler (ранее: **BPwin**) - ведущий инструмент для моделирования бизнес-процессов. Позволяет оптимизировать деятельность организации и проверить ее на соответствие стандартам ISO9000, спроектировать оргструктуру, снизить издержки, исключить ненужные операции и повысить эффективность. Являясь стандартом де-факто, BPwin поддерживает сразу три нотации моделирования: IDEF0 (федеральный стандарт США), IDEF3 и DFD.

AllFusion ERwin Data Modeler

AllFusion ERwin Data Modeler (ранее: **ERwin**) - лидер среди средств моделирования баз данных и хранилищ данных. Позволяет проектировать, документировать и сопровождать базы данных различных типов. Поддерживая прямое и обратное проектирование для 20 типов СУБД, ERwin повышает качество разрабатываемой БД, производительность труда и скорость разработки. Журнал "КомпьютерПресс" по итогам 2000 года признал ERwin лучшим средством проектирования данных.

AllFusion Data Model Validator

AllFusion Data Model Validator (ранее: **ERwin Examiner**) – инструмент для проверки структуры баз данных и создаваемых в ERwin моделей, позволяющий выявлять недочеты и ошибки проектирования. ERwin Examiner дополняет функциональность ERwin, автоматизируя трудоемкую задачу поиска и исправления ошибок, одновременно повышая квалификацию проектировщиков баз данных благодаря встроенной системе обучения.

AllFusion Model Manager

AllFusion Model Manager (ранее: **ModelMart**) - среда для работы группы проектировщиков на ERwin и BPwin. Обеспечивает совместный доступ и редактирование моделей, повышая эффективность и скорость работы проектировщиков, является интегрирующим звеном для ERwin (моделирование баз данных) и BPwin (моделирование бизнес-процессов). Защищает хранимые на собственном сервере модели, позволяя задавать для сотрудников различный уровень доступа к ним. Руководителям же проектов позволяет координировать весь ход работы.

AllFusion Component Modeler

AllFusion Component Modeler (ранее: **Paradigm Plus**) - мощное CASE-средство для моделирования компонентов программного обеспечения и генерации объектного кода

приложений на основе созданных моделей. Продукт можно использовать как при создании новых приложений, так и при изменении или объединении существующих. Благодаря интеграции с BPwin есть возможность использования функциональной модели вместе с объектной. Paradigm Plus поддерживает около десятка стандартных нотаций, таких как UML и Booch, интегрируется с технологиями COM/DCOM, CORBAPlus, Visibroker и др., продуктами CA, Microsoft, Rational Software и др.

2.2 Функциональные возможности CASE-средств

AllFusion Process Modeler (далее BPwin) — CASE-средство для моделирования бизнес-процессов, позволяющая создавать диаграммы в нотации IDEF0, IDEF3, DFD. В процессе моделирования BPwin позволяет переключиться с нотации IDEF0 на любой ветви модели на нотацию IDEF3 или DFD и создать смешанную модель. BPwin поддерживает функционально-стоимостной анализ (ABC).

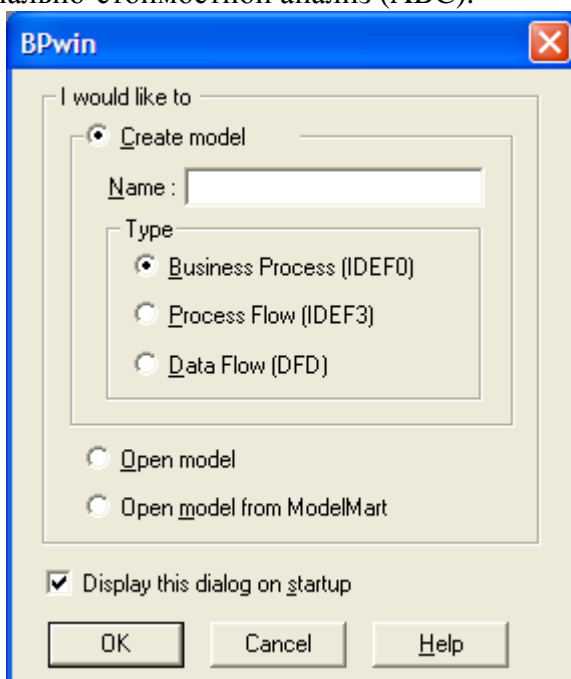


Рисунок 1 – Окно выбора создаваемой новой модели

От выбора типа модели зависит в каких нотациях можно производить декомпозицию работ. Так, если выбрать тип *Business Process (IDEF0)*, то в созданной модели можно производить декомпозицию работ в нотациях IDEF0, IDEF3 и DFD; если выбран тип *Data Flow (DFD)* — в нотациях DFD и IDEF3; если же выбран тип *Process Flow (IDEF3)* — то только в нотации IDEF3.

3 Методические указания по выполнению работы

3.1 Постановка задачи

Изучите материал ЛР и сделайте заключение о достаточности функциональных возможностей данного CASE – средства.

3.2 Анализ функциональных возможностей CASE-средств

Сначала изучите функциональные возможности данного CASE – средства. Затем обоснуйте достаточность или недостаточность набора создаваемых моделей для проектирования ИС.

Основы работы с CASE-средством AllFusion Process Modeler

AllFusion Process Modeler (далее BPwin) — CASE-средство для моделирования бизнес-процессов, позволяющая создавать диаграммы в нотации IDEF0, IDEF3, DFD. В процессе моделирования BPwin позволяет переключиться с нотации IDEF0 на любой ветви модели на нотацию IDEF3 или DFD и создать смешанную модель. BPwin поддерживает функционально-стоимостной анализ (ABC).

Работа с программой начинается с создания новой модели, для которой нужно указать имя и тип (рис.1).

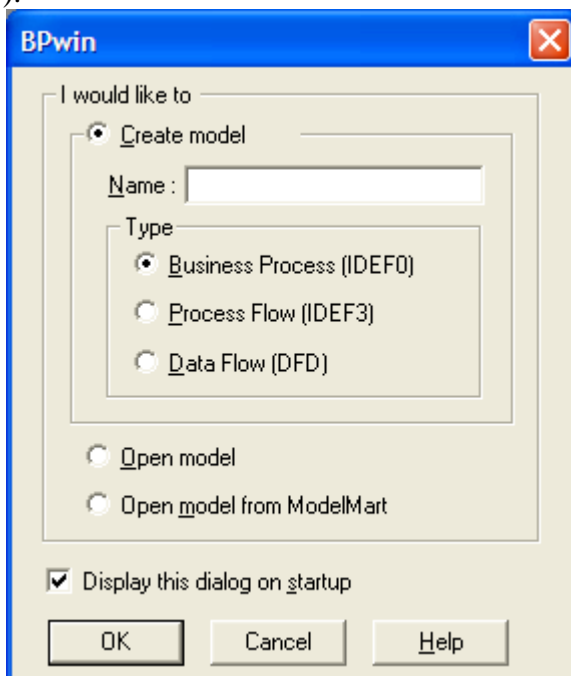


Рисунок 1. Создание новой модели

От выбора типа модели зависит в каких нотациях можно производить декомпозицию работ. Так, если выбрать тип *Business Process (IDEF0)*, то в созданной модели можно производить декомпозицию работ в нотациях IDEF0, IDEF3 и DFD; если выбран тип *Data Flow (DFD)* — в нотациях DFD и IDEF3; если же выбран тип *Process Flow (IDEF3)* — то только в нотации IDEF3.

После ввода имени модели и выбора ее типа BPwin сразу предложит задать параметры модели (рис. 2):

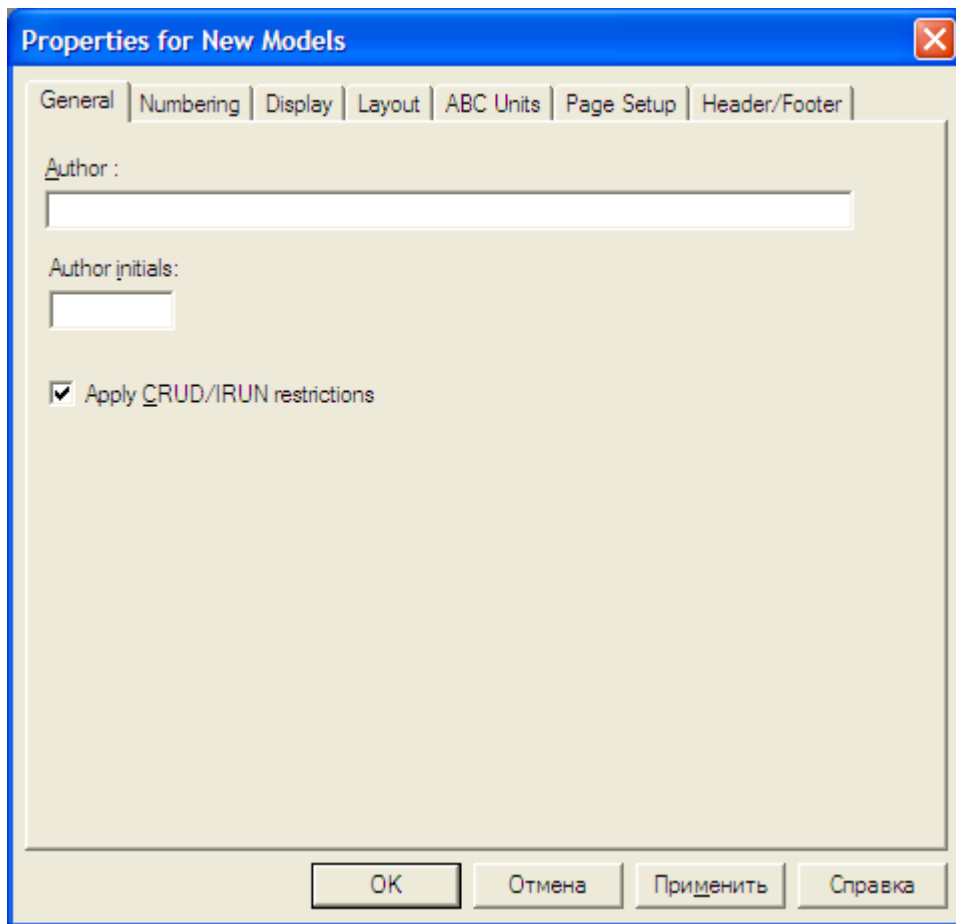


Рисунок 2. Окно задания свойств модели

General— автор модели и его инициалы;

Numbering — формат нумерации работ и диаграмм и порядок ее отображения на диаграммах;

Display — список элементов отображения на диаграммах;

Layout — параметры расположения;

ABC Units — единицы функционально-стоимостного анализа;

Page Setup — параметры страницы;

Header/Footer — параметры верхнего и нижнего колонтитула.

После задания свойств модели появляется главное окно программы (рис. 3), состоящее из трех основных частей:

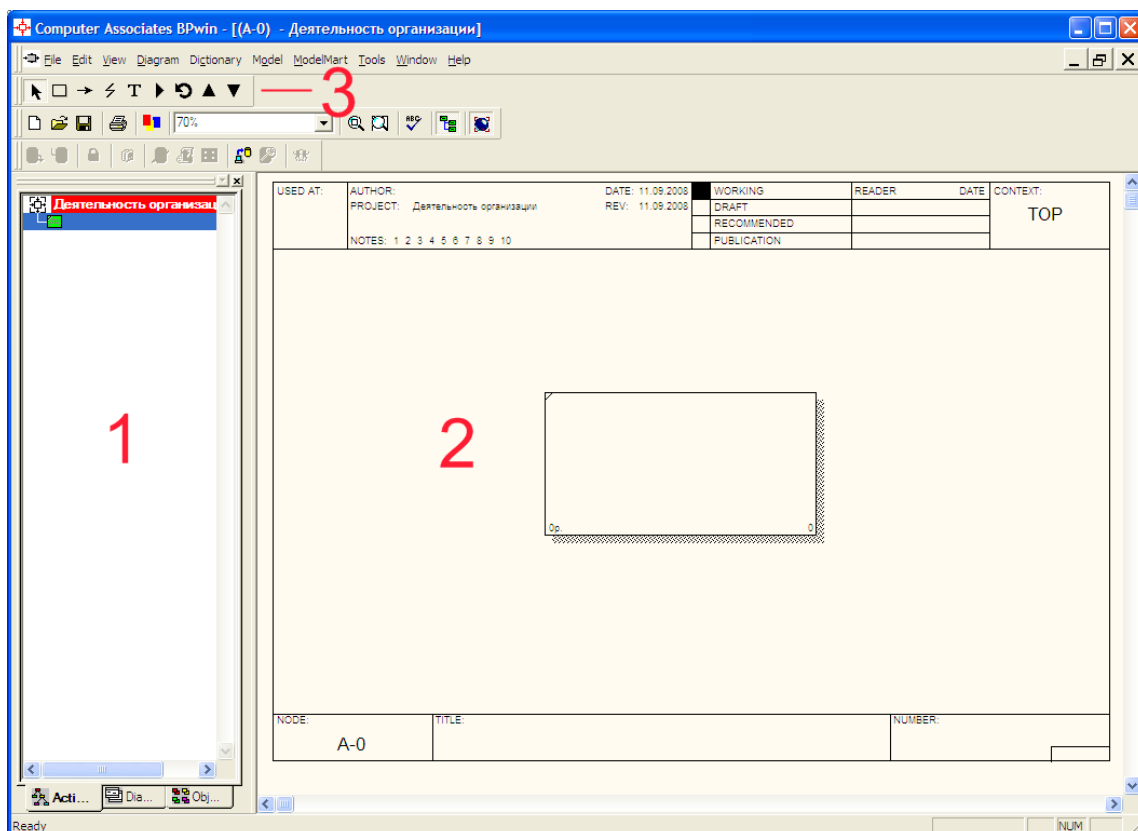


Рисунок 3. Главное окно программы

1 - обозреватель модели (Model Explorer) — отображает структуру модели (имеющиеся диаграммы и их иерархию);

2 - основная часть — в ней отображаются диаграммы, с которыми ведется работа;

3 - панели инструментов, из которых наибольший интерес представляет панель инструментов *Model Toolbox*.

Примечание. В созданной модели с настройками по умолчанию некорректно отображаются русские символы. Чтобы устранить этот недостаток, необходимо подкорректировать используемые в модели шрифты. Для этого в меню *Model -> Default Fonts* необходимо последовательно пройти по всем пунктам (рис. 4), выбрать в выпадающем списке *Script* значение *кириллический* и поставить галочку *Change all occurrences* (рис. 5).

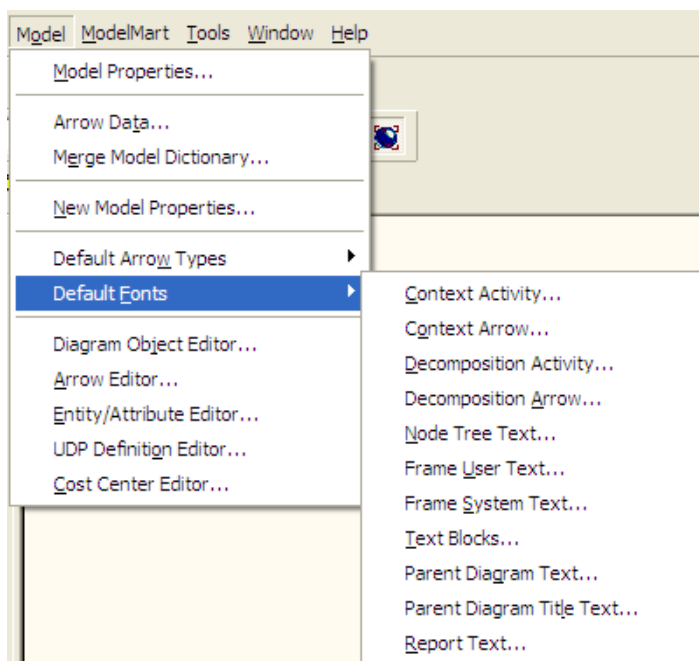


Рисунок 4. Пункты меню, отвечающие за настройки шрифта

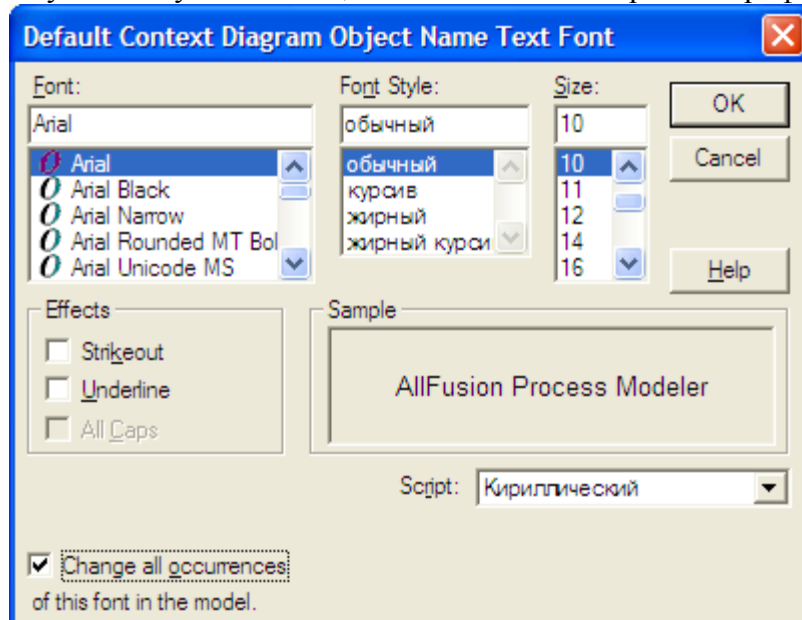












Рисунок 5. Параметры шрифта

Панель инструментов *Model Toolbox*.

Данная панель инструментов отвечает за создание разнообразных графических элементов модели. В зависимости от типа текущей диаграммы набор кнопок на ней меняется.

Таблица 1 - Вид и назначение кнопок Model Toolbox

Вид кнопки	Название кнопки	Назначение кнопки
	Pointer Tool	Превращает курсор в стрелку указателя для того, чтобы можно было выделять объекты
 - IDEF0	Activity Box Tool	Добавление на диаграмму новой работы
 - DFD		

 - IDEF3		
	Precedence Arrow Tool	Добавление на диаграмму новой стрелки
	Squiggle Tool	Связывание названия стрелки с самой стрелкой
	Text Tool	Добавление на диаграмму текста
	Diagram Dictionary Editor	Вызов окна менеджера диаграмм для просмотра имеющихся диаграмм по типам и переход к выбранной
	Go to Sibling Diagram	Переход между стандартной диаграммой, деревом узлов и FEO диаграммой
	Go to Parent Diagram	Переход к родительской диаграмме
	Go to Child Diagram	Переход к дочерней диаграмме
 - DFD	External Reference Tool	Добавление на диаграмму внешней сущности
 - DFD	Data store Tool	Добавление на диаграмму хранилища данных
 - IDEF3	Junction Tool	Добавление на диаграмму перекрестка
 - IDEF3	Referent Tool	Добавление на диаграмму объекта ссылки

Созданная модель уже содержит контекстную диаграмму с единственной работой ("черный ящик") в той нотации, которая была выбрана на этапе создания модели. Теперь необходимо дать этой работе название и при необходимости задать ее свойства. Для этого нужно вызвать окно свойств работы, дважды щелкнуть по ней мышью (рис. 6).

Далее необходимо разместить на диаграмме стрелки. Для этого следует нажать на *Model Toolbox* кнопку *Precedence Arrow Tool* (курсор примет форму крестика со стрелкой), щелкнуть по тому месту, откуда стрелка должна выходить и затем щелкнуть по тому месту, куда стрелка должна заходить (BWin подсветит эти места при наведении на них курсора). Для задания названия стрелки нужно нажать на *Model Toolbox* кнопку *Pointer Tool* и затем дважды щелкнуть по стрелке. В появившемся окне *Arrow Properties* название работы вводится в поле *Arrow Name* или выбирается из списка имеющихся названий стрелок.

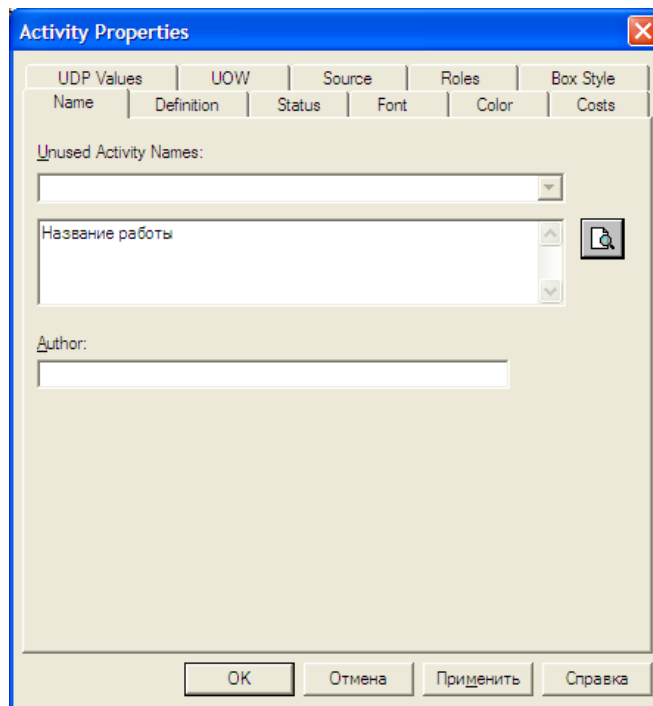


Рисунок 6. Окно свойств работы

После размещения стрелок на диаграмме можно проводить декомпозицию ее работ. Для этого следует нажать на *Model Toolbox* кнопку *Go to Child Diagram* и затем щелкнуть по работе, которую нужно декомпозировать. Появится окно, в котором необходимо выбрать в какой нотации проводить декомпозицию и количество дочерних работ (рис. 7).

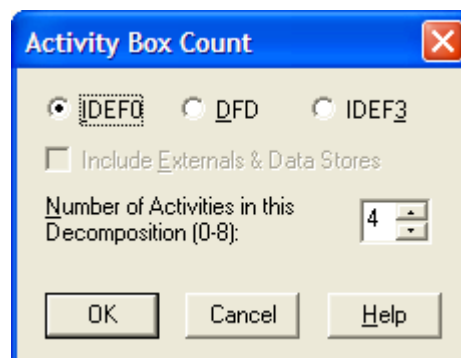


Рисунок 7. Создание дочерней диаграммы

После создания дочерней диаграммы *VRwin* автоматически создаст указанное число работ и разместит граничные стрелки по краям диаграммы. Далее следует связать граничные стрелки со входами работ (при необходимости можно добавить новые граничные стрелки) и связать работы между собой. Дальнейшая декомпозиция работ проводится аналогичным образом.

Контрольные вопросы

- 1 Поясните назначение основных компонентов CASE – средства.
- 2 Какие виды моделей можно создавать с помощью данного CASE – средства?
- 3 Какова обобщенная структура CASE – средства?

Лабораторная работа 2

Анализ возможностей интегрированного комплекса CASE-средств Rational Software Architect

Цель работы:

1. Закрепить основные способы создания CASE-средств.
2. Научиться составлять CASE-средств.

1 Задание на лабораторную работу

- 1.1 Создание CASE-средств
- 1.2 Выполнение CASE-средств

2 Краткие теоретические сведения

2.1 Продукты линейки и их возможности фирмы IBM

Интегрированный набор возможностей для построения систем управления жизненным циклом для различных инженерных дисциплин

Линейка продуктов IBM Rational предоставляет интегрированный набор возможностей для построения систем управления жизненным циклом для различных инженерных дисциплин. Данную линейку отличают современная открытая архитектура по интеграции и управления распределенными данными жизненного цикла, мощными возможностями бизнес аналитики и возможности по централизованному управлению распределенными инженерными данными. Линейка продуктов IBM Rational предоставляет все необходимые возможности для построения системы управления жизненным циклом для следующих инженерных дисциплин:

- Системным инженерам.
- Разработчикам программного обеспечения изделий (встраиваемого ПО).
- Разработчикам автоматизированных систем управления.
- Электронщикам.

В дополнение, продукты линейки могут быть интегрированы с PLM для реализации единого информационного пространства между системными инженерами, конструкторами, электронщиками и разработчиками ПО.

Продукты линейки и их возможности

Линейка продуктов IBM Rational предоставляет широкий спектр возможностей для интегрированного управления данными жизненного цикла.

IBM Rational DOORS	Управление требованиями
IBM Rational Rhapsody	Управление архитектурой технических систем и встраиваемого ПО
IBM Rational TeamConcert	Управление работами, изменениями, исходным кодом, сборками ПО
IBM Rational Quality Manager	Управление испытаниями
IBM Rational Software Architect	Управление архитектурой ИТ систем и приложений
IBM Rational Asset Manager	Управление архивом
IBM Rational Insight	Бизнес аналитика на основе данных жизненного цикла
IBM Rational RELM	Управление распределенными инженерными данными
IBM Rational Focal Point	Управление портфелями
IBM Rational System	Управление архитектурой предприятия

Architect	
IBM Rational Publishing Engine	Публикация документов и отчетов

Более подробную информацию о продукте Вы можете найти на [сайте производителя](#).

2.2 Состав CASE-средств Rational Software Architect

Структура системы программных продуктов корпорации IBM представлена на рисунке 1.

WebSphere Платформа для создания прикладных систем	DB2 Система управления базами данных и хранения информации	Lotus Управление документами, коммуникации, поддержка совместной работы	Tivoli Управление ИТ-инфраструктурой масштаба предприятия
Software Development Platform (Rational) Средства разработки и поддержки ПО на протяжении всего жизненного цикла приложений			
Рисунок 1 - Структура системы программных продуктов корпорации IBM.			

В рамках реализации проекта Atlantic фирмой **IBM** представлены четыре новых продукта.

IBM Rational Portfolio Manager - решение для всего жизненного цикла, предназначенное для формирования приоритетов, планирования, управления и оценки ИТ-проекта с использованием методов управления ИТ-ресурсами как бизнесом.

IBM Rational Software Architect - инструмент проектирования и разработки, позволяющий определять и сопровождать все архитектурные аспекты прикладного ПО. Он обеспечивает разработку приложений на языках C/C++, в том числе с использованием компонентов Eclipse. Этот инструмент должен повысить качество ПО за счет проверки архитектуры приложения на соответствие проектным требованиям, устойчивость и возможность внесения изменений.

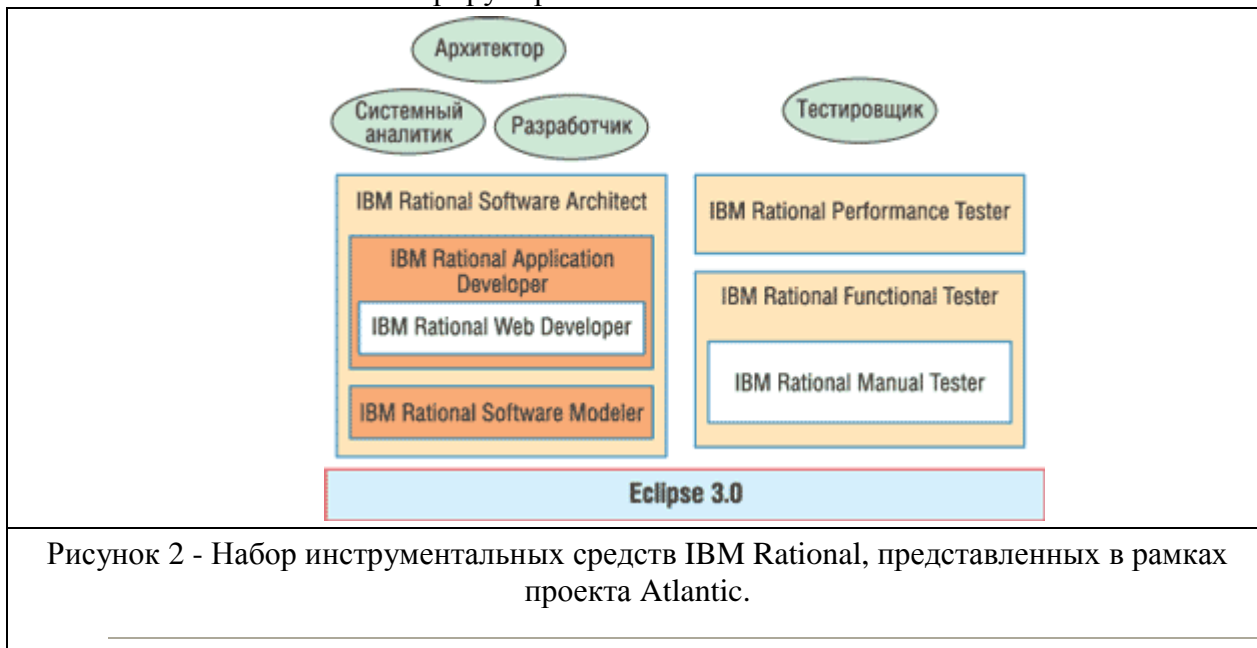
IBM Rational Software Modeler - средство визуального моделирования и проектирования на базе языка UML 2.0, позволяющее организовать документирование и обмен информацией между различными представлениями разрабатываемой системы. С его помощью архитекторы, системные аналитики и проектировщики смогут обеспечить тщательную проработку спецификаций, архитектуры и проекта, а также последующее доведение этой информации до всех участников процесса разработки. В этом инструменте используются самые последние версии Eclipse и UML, что, к примеру, позволяет реализовать такое качественное новшество, как поддержка трансформации моделей.

IBM Rational Manual Tester - средство ручного тестирования, предназначенное для бизнес-аналитиков и тестировщиков. Применяемые в нем технологии направлены на то, чтобы снизить влияние вносимых изменений при создании и применении ручных тестов.

Одновременно был представлен еще ряд обновленных продуктов для разработчиков ПО - усовершенствованные средства тестирования функциональности и производительности (IBM Rational Functional Tester и IBM Rational Performance Tester) и

улучшенные инструменты для управления конфигурациями ПО (SCM-решения), включая IBM Rational ClearCase и IBM Rational ClearQuest.

Все новые инструменты построены на базе Eclipse 3.0, их местоположение в общей системе и взаимосвязи иллюстрирует рис. 2.



IBM® Rational® Software Architect (часть IBM Software Development Platform) - это первый значительный релиз продуктов Rational от IBM, начиная с февраля 2003 года. Разработчикам, работающим с продуктами от IBM, понравятся улучшенные возможности Software Development Platform: новый уровень интеграции и удобство в использовании.

Software Development Platform состоит из нескольких продуктов от IBM Rational:

- **IBM® Rational® Portfolio Manager** это инструмент для управления проектами, который включает в себя весь жизненный цикл проекта. Rational Portfolio Manager помогает отслеживать портфельные инвестиции в реальном времени. Это в свою очередь позволяет менеджерам принимать решения, будучи более информированными. Этот продукт можно также использовать для увеличения сотрудничества между различными группами, работающими над проектом.

- **IBM® Rational® Application Developer и Rational Web Developer:** являются ребрендингами соответственно IBM® WebSphere® Studio Application Developer и IBM® WebSphere® Studio Site Developer интегрированных сред разработки (IDE). Эти продукты являются базисом, на котором построены IBM® Rational® Software Modeler и IBM Rational Software Architect.

- **IBM Rational Software Modeler: Rational Software Modeler** - это инструмент, основанный на Unified Modeling Language (UML) 2.0, что позволяет архитекторам и аналитикам создавать различные виды системы. Этот инструмент стимулирует эффективное сообщение между архитекторами и дизайнерами с одной стороны и разработчиками - с другой.

- **IBM Rational Software Architect:** Этот инструмент позволяет архитекторам создавать и сохранять архитектуру приложений. Rational Software Architect будет уделено большое внимание в этой статье.

- **IBM® Rational® Manual Tester:** Этот инструмент позволит аналитикам и тестировщикам создавать и реализовывать тестовые примеры. Rational Manual Tester предназначен для тестирования.

- **IBM® Rational® Functional Tester:** Этот инструмент можно использовать для автоматизации функциональных и регрессивных тестов.

- **IBM® Rational® Performance Tester:** Этот инструмент может автоматизировать загрузку и выполнение тестов для ваших приложений.

Благодаря тому, что Rational Software Architect является совокупностью Rational Application Developer, Rational Web Developer и Rational Software Modeler, вы можете получить все эти инструменты, скачав Rational Software Architect. За дополнительной информацией о них посетите сайт IBM's Rational.

2.3 Функциональные возможности CASE-средств

Rational Software Architect - инструмент, обеспечивающий следующие возможности:

- Моделирования на основе UML
- Инфраструктуры шаблонов
- Функций трансформации модели и генерации кода
- Инструмента проектирования и разработки для конкретной платформы, а также среды для тестирования элементов.

Это интегрированное средство проектирования и разработки, которое использует преимущества UML-разработки на основе моделей, позволяя создавать приложения и сервисы с практичной архитектурой. С Rational Software Architect можно:

- Усилить открытую и расширяемую платформу моделирования
- Ускорить моделирование и проектирование ПО
- Автоматизировать процессы разработки и максимизировать многократное использование ресурсов
- Повысить производительность при разработке приложений и Web-сервисов

IBM® Rational® Software Architect не только дает возможность поддержки сквозной (end-to-end) разработки под управлением моделей (MDD), но и дополнительные функции для поддержки разработки на основе ресурсов, которые дополняют MDD за счет поддержки многократного использования моделей, шаблонов и трансформаций, как показано на рисунке 4.

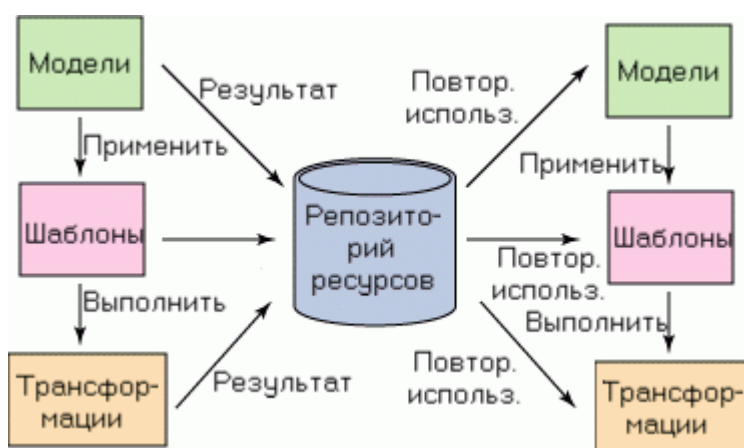


Рисунок 4 - Использование моделей, шаблонов и трансформаций

2.4 Применение CASE-средств

Разработка ПО, управляемая моделями (MDD), - это новая парадигма разработки программного обеспечения, управляемая и поддерживаемая методами архитектуры Model-driven Architecture (MDA) (подход к разработке ПО, созданный командой Object Management Group (OMG)).

Одна из главных особенностей MDD - использование моделей в качестве ключевых артефактов. Модель - это описание системы с определенной точки зрения, с опущением несущественных деталей, с тем чтобы более четко выделить характеристики, представляющие интерес. В MDD модель должна быть машинно-читаемой, чтобы обеспечить автоматический доступ к ее содержанию, а также генерацию артефактов. Диаграмма на демонстрационной доске может соответствовать остальным критериям, необходимым, чтобы считаться моделью, но только зафиксировав ее в машинно-читаемой форме, вы сможете использовать ее в системе инструментов MDD.

Рисунок 1 демонстрирует, как с помощью MDD коммерческая проблема преобразовывается в решение ПО. Бизнес-проблема анализируется, а затем применяются общие бизнес-шаблоны. Таким образом, модель разработки частично заполняется, в ней появляются детали конкретной разрабатываемой бизнес-функции. Далее внедряются независимые от платформы шаблоны разработки для трансформации модели разработки в ряд независимых от рабочего цикла компонентов. После этого выбирается исполняемая платформа и специфические для ее рабочего цикла шаблоны применяются для генерации артефактов.

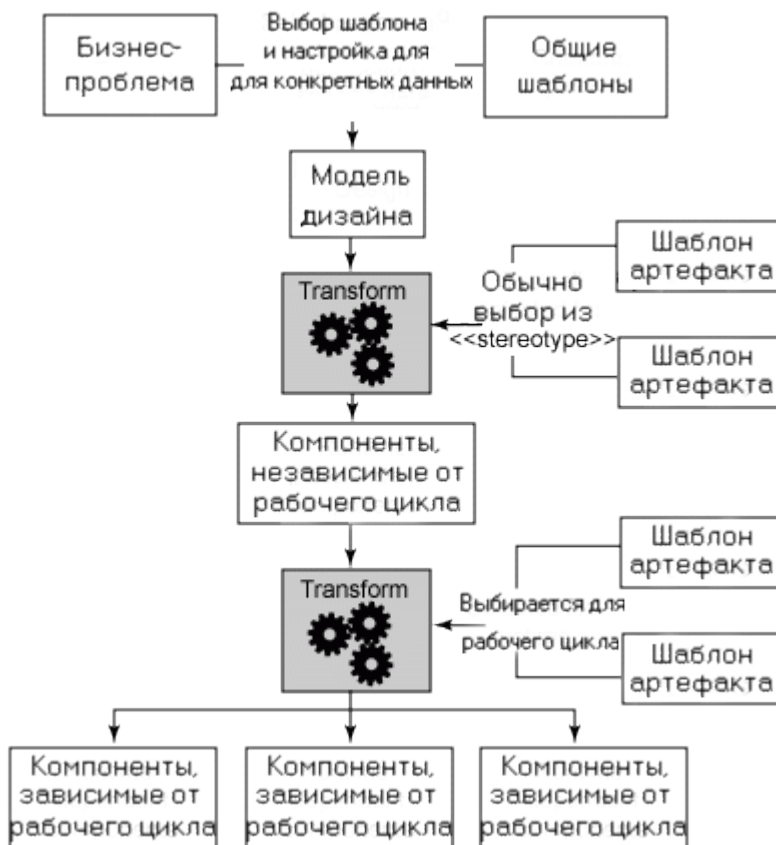


Рисунок 5 - Использование MDD для преобразования бизнес-проблемы в программное решение

Модели, шаблоны и трансформации являются ресурсами многократного использования. После разработки их обрабатывают, организуют, хранят и используют для ускорения MDD и для обеспечения наибольшей выгоды от затраченных усилий. В результате этого разработка ресурсов становится важной частью MDD.

Rational Software Architect применяет спецификацию ресурсов многократного использования Reusable Resource Specification (RAS) в качестве стандартного способа упаковки и использования ресурсов, которые обеспечивают решение проблемы в конкретном контексте. RAS-ресурс представляет собой коллекцию взаимосвязанных артефактов с декларацией (manifest), которая описывает этот ресурс, что показано на Рисунке 10. **Артефакты** — любые созданные из жизненного цикла ПО, например,

документация по требованиям, модели, файлы с исходным кодом, дескрипторы развертывания, примеры тестирования и скрипты. Как правило, термин *артефакт* применяют к файлу.

В Rational Software Architect входят функции, которые помогут показать архитектурное построение вашей модели или кода, что поможет вам понять вашу модель с точки зрения архитектора

Например, архитектурный анализ представляет собой набор мероприятий по созданию и совершенствованию архитектуры программного обеспечения системы. При итеративном использовании такое обдумывание архитектуры помогает ставить и решать задачи при разработке программного обеспечения, не требуя значительных усилий по предварительному созданию архитектуры. Архитектурный анализ имеет решающее значение для любой сложной программной системы. Технология итеративного и поэтапного архитектурного анализа может быть представлена в виде рисунка 6.



Рисунок 6 - Итеративный и поэтапный архитектурный анализ

RAS-ресурсы можно хранить в репозиториях ресурсов на основе RAS. При упаковывании ресурса в Rational Software Architect, вы можете экспортировать его в соответствующий репозиторий ресурсов. Можно также опубликовать ресурс в файле формата .gas из файловой системы в связанный репозиторий из представления Asset Explorer в перспективе RAS.

Кроме того, есть возможность просматривать, осуществлять поиск и импортировать ресурсы из репозитория через Asset Explorer, как показывает Рисунок 8. Rational Software Architect поддерживает следующие 4 типа репозитория ресурсов на основе RAS. Можно задавать связи с этими репозиториями из представления Asset Explorer.

Локальный репозиторий: Репозиторий для локальной файловой системы для индивидуального пользования. С командой этим репозиторием пользоваться совместно нельзя, если только вы не используете сетевую файловую систему для хостинга локального репозитория.

Репозиторий рабочей группы: Репозиторий на основе Web-сервиса для пользования членов команды. Это приложение к Java 2 Platform, Enterprise Edition (J2EE), которое устанавливается на сервер приложений J2EE. Впоследствии его можно использовать для совместного использования ресурсов среди членов команды, у которых установлен Rational Software Architect. Загрузить Репозиторий рабочей группы можно с сайта IBM® alphaWorks.

XDE-репозиторий: Эта опция обеспечивает поддержку старой версии репозитория Rational XDE.

developerWorks-репозиторий: Он находится на сайте IBM® developerWorks®. Можно задавать поиск и импорт новых шаблонов дизайна, которые создаются в IBM. Число шаблонов постоянно растет.

В настоящее время IBM® Rational® Software Modeler и IBM Rational Software Architect Standard Edition теперь объединены в один основной продукт: Rational Software Architect.

Версия 8.0 представляет собой основной релиз семейства продуктов IBM® Rational® Software Architect. Программный пакет Rational Software Architect был собран заново в виде продукта базового уровня, который можно расширять в соответствии с необходимыми предметно-ориентированными возможностями.

Он предоставляет (по значительно более низкой цене в сравнении с предыдущим продуктом Rational Software Architect) нотацию BPMN 2 (Business Process Modeling Notation), моделирование на UML 2 (Unified Modeling Language), визуализацию кода и согласованную поддержку моделирования для Java, C# и VB.NET (Microsoft Visual Basic.NET). Эту базовую платформу можно наращивать при помощи набора расширений, которые обеспечивают дополнительные возможности, от совместной работы и имитационного моделирования до моделирования развертывания и использования интегрированных архитектурных сред.

Логическая схема Rational Software Architect и его расширений, отражающая функциональные возможности версия 8.0, показана на рисунке 3

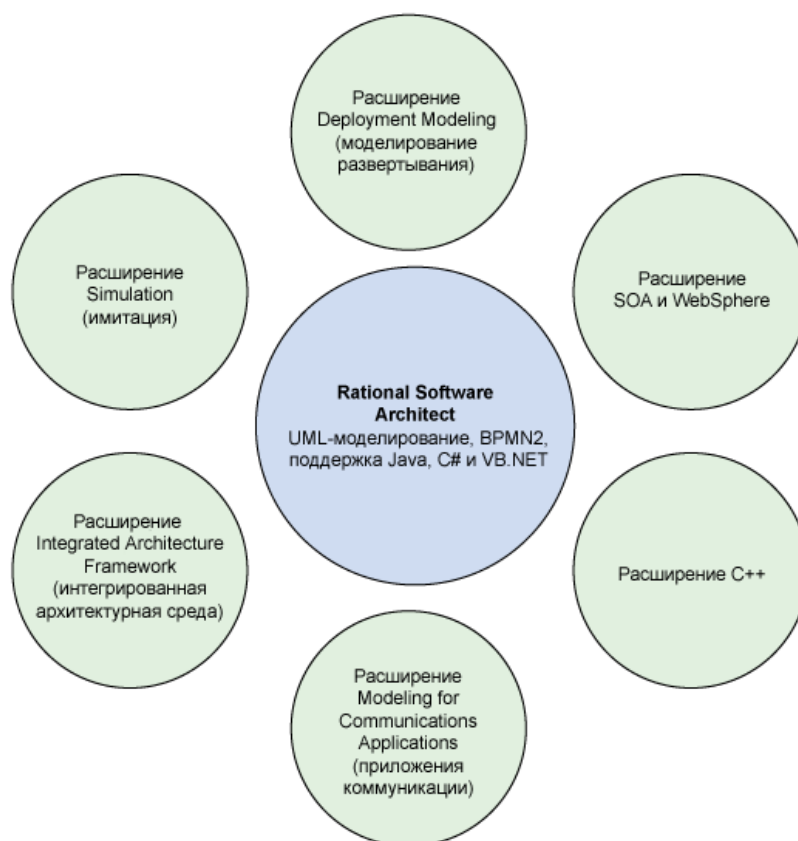


Рисунок 3 - Логическая схема Rational Software Architect и расширений

Расширение Simulation (имитационное моделирование)

При помощи расширения Simulation можно имитировать любое UML-поведение (диаграммы деятельности, диаграммы последовательности, диаграммы коммуникации или диаграммы состояний).

Имитация может работать как на неформальных моделях, так и на формальных, для

которых с целью создания строгих спецификаций был применен язык UML Action Language (UAL).

Расширение Deployment Modeling (моделирование развертывания)

Расширение Deployment Modeling в настоящее время включает в себя общие улучшения моделирования, которые были добавлены к UML-моделям, такие как темы, слои и обратная связь в диаграммах. Это означает, что теперь стало намного проще создавать и понимать сложные топологии развертывания. Также были выполнены значительные улучшения в определении собственных элементов топологии, интеграции с UML-моделями, поддержке типов развертывания и использовании топологии для руководства автоматизацией.

Расширение Deployment Modeling теперь поддерживает следующие архитектуры Microsoft®: Internet Information Services (IIS), SQL Server®, .NET® Framework, Silverlight®, ASP.NET, Windows Communication Format (WCF), Windows Presentation Format (WPF) и многие другие.

Расширение C++

В Rational Software Architect V8 расширение C++ предоставляет три стандартных способа связать модель с кодом при коллективной разработке на C++:

Визуализация, способствующая пониманию существующего кода.

Смешанное моделирование для менее сложных систем и групп, придерживающихся методики agile-разработки.

Согласованное моделирование в сложных системах, где важно поддерживать абстракции в модели.

Расширение Modeling for Communications Applications (моделирование коммуникационных приложений)

Расширение Modeling for Communications Applications ускоряет разработку приложений конвергентных сервисов сетей Next Generation Networks (NGN), поддерживая:

проектирование и создание сервлетов Session Initiation Protocol (SIP);

проектирование потоков вызовов SIP;

создание тестов SIPp для утилиты тестирования с открытым исходным кодом SIPp;

готовые типовые модели и руководства New Generation Operations Systems Software (NGOSS);

базовую SoaML-модель ParlayX (набор стандартных [API web-сервисов](#) для [эксплуатационных](#) и биллинговых служб телекоммуникационной индустрии) и возможность создания телекоммуникационных web-сервисов на основе ParlayX.

Rational Software Architect v8 обеспечивает поддержку SIP 1.1, а также возможность преобразования потоков вызовов в SIPP и SIPP в потоки вызовов.

SOA-преобразования

Rational Software Architect поддерживает концепцию модели сервисов посредством SoaML. В версии 8 можно выполнять преобразования из различных источников, таких как IBM WebSphere Business Modeler, EJB (Enterprise JavaBeans) или WSDL для создания новых и обновления имеющихся представлений ваших корпоративных сервисов и их зависимостей. Преобразование модели сервиса позволяет создавать (и объединять существующие) артефакты кода, XML, WSDL, BPEL (Business Process Execution Language) и Service Component Architecture (SCA). Результаты преобразования можно, например, использовать для оркестровки сервисов в таких инструментах, как IBM WebSphere Integration Developer.

Расширение SOA и WebSphere

Версия 8 предоставляет полный комплекс возможностей сервис-ориентированной архитектуры (SOA) и WebSphere в рамках отдельного продукта, который включает в себя весь IBM® Rational® Application Developer плюс SOA-преобразования, а также новую возможность создания SoaML-модели сервисов из имеющейся WSDL-модели (Web Services Description Language). Это позволяет быстро построить модель текущего

состояния вашего бизнеса, используя ее, перейти к моделированию новых сервисов и потоков бизнес-процессов, а затем на основе этих моделей создать код необходимых сервисов. Мы также добавили поддержку последней версии спецификации SoaML 1.0 Beta 2 и упростили использование и повторное использование BPMN2-моделей процессов для запуска и определения моделей сервисов.

Расширение Integrated Architecture Framework (интегрированная архитектурная среда)

Расширение Integrated Architecture Framework обеспечивает поддержку моделирования взаимосвязей архитектурных решений с бизнесом (организация, стратегия, инициативы, цели и т.д.) при помощи IBM UPIA (UML Profile for Integrated Architectures).

3 Методические указания по выполнению работы

3.1 Постановка задачи

Изучите материал ЛР и сделайте заключение о достаточности функциональных возможностей данного CASE – средства.

3.2 Анализ функциональных возможностей CASE-средств

Сначала изучите функциональные возможности данного CASE – средства. Затем обоснуйте достаточность или недостаточность набора создаваемых моделей для проектирования ИС.

Контрольные вопросы

- 1 Поясните назначение основных компонентов CASE – средства.
- 2 Какие виды моделей можно создавать с помощью данного CASE – средства?
- 3 Какова обобщенная структура CASE – средства?

Лабораторная работа 3

Анализ возможностей интегрированного комплекса CASE-средств Rational Software Architect

Цель работы:

1. Закрепить основные способы создания CASE-средств.
2. Научиться составлять CASE-средств.

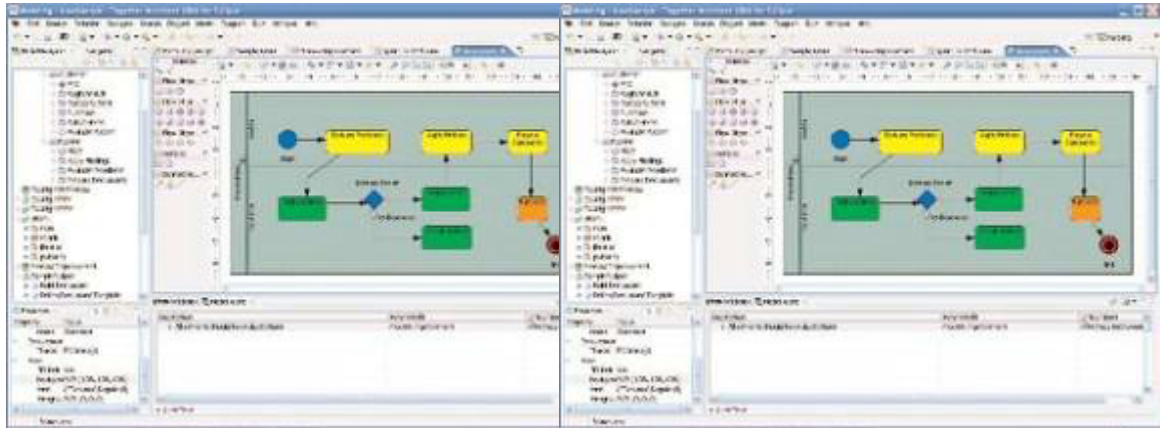
1 Задание на лабораторную работу

- 1.1 Создание CASE-средств
- 1.2 Выполнение CASE-средств

2 Краткие теоретические сведения

2.1 Состав CASE-средств

Borland Together



Borland Together - семейство продуктов, предназначенных для визуального моделирования и проектирования программных систем на основе стандарта UML.

-
- **Варианты поставки Borland Together**
-
- **Borland Together ControlCenter**
- Together ControlCenter объединяет этапы дизайна, разработки и развертывания приложений в единый комплекс, что упрощает и повышает степень интеграции анализа, дизайна, реализации, развертывания и отладку комплекса приложений.
- **Поддержка нескольких языков программирования.** Уникальная платформа Borland Together ControlCenter поддерживает следующие языки программирования: Java, C++, CORBA , IDL, Microsoft Visual Basic 6, Visual Basic .NET и C# – все языки в одном продукте. Поддержка нескольких языков программирования способствует повышению рентабельности предназначенного для разработки программного обеспечения решения, поскольку группы разработчиков могут использовать его, программируя сразу на нескольких языках.
- **Развертывание на разных серверах приложений.** ControlCenter позволяет выбирать для развертывания различные серверы приложений, включая многочисленные версии таких серверов, как BEA WebLogic Server и IBM WebSphere, а также Sun One, Sun EE Reference Implementation, SAP Application Server, Jboss 2.0 и Oracle9i. Кроме того, ControlCenter значительно облегчает процесс развертывания и реорганизации проектов на нескольких различных серверах.
- **Создание сценариев тестирования.** Автоматизация разработки сценариев тестирования для блочной и визуальной проверки программ, использующих библиотеки Swing and AWT. Функциональные средства обеспечения качества в ControlCenter улучшают качество кода и позволяют группам разработчиков повышать качество программного обеспечения в ходе разработки.
- **Интеграция с передовыми технологиями.** ControlCenter интегрируется не только с решениями Borland, но и с лучшими программными продуктами других компаний, а также со всеми SCC-совместимыми системами контроля версий, включая Borland StarTeam и Rational ClearCase и с ведущими средствами управления требованиями, такими как Borland CaliberRM и Rational RequisitePro. Возможность интеграции ControlCenter с передовыми технологиями позволяет разработчикам выбирать для своих проектов лучшие из всего спектра программных решений.
- **Borland Together Solo**

- Borland Together Solo посредством UML-инструментов дизайна и разработки обеспечивает небольшие команды и работающих индивидуально программистов всем необходимым для быстрого и эффективного создания приложений небольшого объема.
- **Borland Together Edition for JBuilder**
- Borland Together Edition for JBuilder – это интегрированная среда быстрой разработки приложений, позволяющая командам проектировщиков и программистов ускорить разработку высокопроизводительных приложений, используя JBuilder – самый распространенный Java-продукт. Интеграция JBuilder с технологиями Together позволяет проводить проверку синхронизации моделей и кода и обеспечивает идентичной информацией всех сотрудников группы разработки. Кроме того, связывание таких инструментов обеспечения качества, как аудит и метрики, позволяет пользователям широко использовать комбинацию JBuilder с Together для быстрого выявления и исправления ошибок в коде. Интеграция JBuilder с технологиями Together не только экономит время за счет объединения этих средств на серверах, но и позволяет пользователям получить опыт комплексного решения.
- **Borland Together Edition for Microsoft Visual Studio .NET**
- Использование интегрированной среды быстрой разработки приложений Borland Together Edition for Microsoft Visual Studio .NET поможет вам оперативно создавать высококачественные приложения с использованием платформы Visual Studio .NET
- **Borland Together Edition for C++BuilderX**
- Использование интегрированной среды быстрой разработки приложений Borland Together Edition for C++BuilderX поможет вам оперативно создавать высококачественные приложения с использованием многоплатформенной среды разработки C++Builder
- **Borland Together Edition for WebSphere Studio**
- Расширяет функциональность IBM WebSphere Studio, позволяя поддерживать моделирование, разработку и развертывание проектов программного обеспечения в среде WebSphere Studio. Инструмент Borland Together Edition for WebSphere Studio был разработан в сотрудничестве с IBM специально для нужд разработчиков Java и Web-приложений, работающих со средой WebSphere Studio Application Developer. Together Edition for WebSphere Studio расширяет функциональные возможности среды IBM WebSphere Studio в моделировании и создании и развертывании проектов по разработке программного обеспечения, выполняемых исключительно в этой среде. Together Edition for WebSphere Studio обеспечивает пользователей передовой технологией Together для среды IBM.
- **Borland Together Edition for Eclipse**
- Интеграция технологий Together с отраслевой платформой Eclipse с открытым исходным кодом предлагает пользователям Eclipse хорошо известную среду разработки программного обеспечения. Теперь пользователи среды Eclipse могут использовать передовые средства моделирования, проектирования и обеспечения качества, входящие в пакет Together Edition for Eclipse, для скорейшей разработки высококачественных корпоративных приложений. Тесная интеграция обеих технологий в одном средстве предлагает уникальные возможности для разработки пользовательского интерфейса в среде Eclipse для Together Edition for Eclipse.
- **Borland Together Edition for SAP NetWeaver Studio**
- Borland Together Edition for SAP NetWeaver Studio – это многократно завоевывавшая различные награды среда моделирования тесно интегрированная со средой разработки SAP NetWeaver.
- **Together Designer Community Edition**
- Borland Together Designer Community Edition - это среда моделирования, предназначенная для создания Unified Modeling Language (UML) 2.0 диаграм.

Borland Together Designer Community Edition предлагает возможность моделирования приложений и представления архитектуры приложений в виде точных, легковоспринимаемых изображений, облегчая тем самым понимание и взаимодействие разработчиков при проектировании архитектуры систем.

2.2 Функциональные возможности CASE-средств

Возможности и преимущества:

Решения в области моделирования для бизнес-аналитиков, системных аналитиков, архитекторов и разработчиков

Borland Together призван обеспечить соответствие специальным требованиям к моделированию всех ролей в коллективе разработчиков ПО. Это позволит участникам группы эффективно взаимодействовать для создания высококачественных приложений за меньшее время. Коллективы, работающие над новыми или существующими бизнес-процессами, анализом, дизайном и архитектурой, выигрывают от расширенных возможностей обмена информацией и снижения рисков задержки проекта.

Значительное повышение гибкости бизнеса и снижение затрат на сопровождение приложений с помощью предметно-ориентированных языков

Модели предоставляют проект для бизнес-процесса, приложения и корпоративных архитектур, а также для структур данных. Эти модели необходимы для обмена информацией между проектными группами и обеспечивают надежность архитектуры. Визуализация моделей помогает организациям справляться со сложностью задач, повышая широту охвата, информативность и ценность документации. Поскольку сложность приложений и бизнес-процессов растет, то также возрастает и важность хороших методик моделирования в обеспечении корректности, качества, и, что не менее важно, возможности долгосрочной поддержки. Для снижения общей величины совокупной стоимости владения (ТСО) в ИТ-организациях очень важны оптимизация бизнес-процессов, дизайн приложений и методики создания решений на базе предыдущего опыта.

Также растет важность возможности использования моделей, которые определяют предметно-ориентированные языки (domain-specific language, DSL). Это обеспечивает план в контексте бизнеса. Предметно-ориентированные языки можно дополнять визуализацией, интеграцией на основе моделей, преобразованием и возможностями генерации моделей, которые предоставляют дополнительные выгоды для бизнеса. Это устраняет общую сложность моделирования и позволяет группам разрабатывать модели в пределах своей отдельной бизнес-области для оптимальной эффективности и обмена информацией.

Первоклассные шаблоны проектирования позволяют многократно добиваться успехов в проектах

Borland Together дает группам разработчиков ПО возможность создавать и многократно использовать проверенные стандартные шаблоны проектирования для определенной отрасли. Это гарантирует создание высококачественных приложений и способствует использованию успешных проектов. Эффективность работы коллектива повышается, а необходимость повторной переработки из-за ошибок проектирования в жизненном цикле разработки сокращается.

Технологии Together предоставляют передовые возможности проектирования, которые обеспечивают визуализацию ИТ-архитектур.

Платформено-независимое моделирование для различных платформ

Borland Together дает организациям необходимую гибкость для создания платформено-нейтральных дизайнов, которые могут работать на различных платформах. Также Borland Together поддерживает широкий спектр языков программирования, что позволяет архитекторам преобразовывать свои дизайны в модели для конкретной платформы.

Поддержка отраслевых стандартов

Borland Together соответствуют стандартам MDA. Они поддерживают унифицированный язык моделирования (Unified Modeling Language: UML), обмен метаданными XML (XML Metadata Interchange, XMI), запрос, представление, **преобразование** (Query/ Views/Transformations, QVT) и язык объектных ограничений (Object Constraint Language, OCL). Borland оказывает влияние на архитектуру, управляемую моделями (Model Driven Architecture MDA), в частности, на QVT, посредством участия в разработке технологии преобразования моделей. Сегодня эта технология позволяет разработчикам архитектуры преобразовывать любые типы моделей на базе инфраструктуры моделирования Eclipse (Eclipse Modeling Framework, EMF) в другую модель (например, CIM в PIM, PIM в PSM и наоборот).

Эффективная экономия времени на протяжении всего жизненного цикла разработки

Всесторонняя автоматизация и возможности для экономии времени позволяют коллективам разработчиков работать более продуктивно. В число ключевых возможностей входит автоматическая генерация документов, многократное использование программных ресурсов (например, шаблонов и определений компонентов), быстрое распространение изменений через рефакторинг, а также уникальная технология

LiveSource, которая предлагает двунаправленную технологию и синхронизацию моделей и кода в любой момент времени на протяжении всего жизненного цикла разработки.

Основные возможности:

Новое: набор средств для работы с предметно-ориентированными языками (DSL).

- Визуальное моделирование метамodelей для определенной предметной области.
- Создание решений на базе DSL с помощью мастера. В число возможностей входят преобразование моделей в редакторах диаграмм, генераторы кода, а также отчеты BIRT (Business Intelligence and Reporting Tools Project, проект в области бизнес-аналитики и инструментов генерации отчетов) в рамках Eclipse.

- **Диаграммы GMF UML 2.1.**

Моделирование бизнес-процессов.

- Нотация моделирования бизнес-процессов (Business Process Modeling, BPMN) с проверкой корректности.
- Импорт/Экспорт BPEL для веб-сервисов (BPEL4WS).

UML-моделирование.

- Независимое от языка создание диаграмм UML 1.4 и UML 2.0.
- UML-моделирование при помощи LiveSource для Java™/C++/CORBA IDL.
- Дифференциация и слияние моделей.
- Многоязычная поддержка.

Моделирование данных.

- Логическое моделирование данных с помощью профиля UML 2.0 для моделирования данных.
- Физическое моделирование данных при помощи диаграмм отношений сущностей (ER) и IDEF1x.
- Прямое и обратное проектирование для ведущих СУБД (Oracle, DB2, Sybase, MSSQLServer).
- Преобразование модели данных из логической в физическую.

Расширенные возможности моделирования и управляемой моделями архитектуры (MDA).

- Поддержка языка объектных ограничений (ObjectConstraint Language, OCL) версии 2.0, включая подсветку синтаксиса, проверку корректности и стандарт распознавания ввода (code sense).
- QVT для преобразований из модели в модель (OMG).

- Преобразования "модель-текст" при помощи xPand, JET и EMF API.
- Генераторы кода для Java, J2EE, C++ и C# (новая функция).
- Конструирование, применение и развертывание UML-профиля, реализованное в виде подключаемого модуля Eclipse.
- Шаблоны разработки, в том числе поддержка шаблонов группы Gang of Four.
- Распознавание шаблонов проекта исходного кода.
- Проектирование и многократное использование шаблонов кода.
- Импорт и экспорт моделей XMI 2.0.
- Импорт моделей Rose и XDE.

Создание документации.

- Генерирование документации для HTML-портала при помощи навигационного апплета, схем с гиперссылками и отчетов модель/код в стиле Javadoc.
- Создание файлов изображений из диаграмм в нескольких форматах.
- Конструктор шаблонов для настройки документации, макетирование диаграмм для печати, автоматическая генерация документов с использованием параметров командной строки.

Обеспечение качества.

- Аудит кода и метрики.
- Проверка и сбор метрик моделей на базе языка OCL.

Коллективная работа.

- Коллективная работа: совместное использование диаграмм и моделей между проектами с управлением версиями.
- Интеграция с Borland StarTeam.
- Генерация схем контрольных примеров из требований, а также трассирование элементов моделей от требований и к требованиям при помощи программ Borland CaliberRM и RequisitePro.

- Импорт требований UML-моделей из Borland Caliber DefineIT

Платформа.

- Eclipse 3.4.

Системные требования

Операционная система

- Windows Vista, Windows XP Pro (SP 2).
- RedHat Enterprise Linux 4 update 2, x86-32.
- SUSE Linux Enterprise Server 10, x86-32.

- Solaris 10 SPARC, GTK.
- Mac OS X 10. 5, Universal, Carbon.

Память

- Минимум 1 Гб (рекомендуется больший объем памяти). Для достижения оптимальной производительности в крупномасштабных проектах, возможно, потребуется изменить параметры кучи (heap) виртуальной Java-машины (JVM) в файлах bat, .cmd или .sh, расположенных в подкаталоге bin установочного каталога Together. Дополнительную информацию можно найти в часто задаваемых вопросах на веб-сайте компании Sun по адресу: [/go.asp?url=-3D-41-52-17-22-18-3E-29-DD-A2-99-6F](http://go.asp?url=-3D-41-52-17-22-18-3E-29-DD-A2-99-6F) sun.com/docs/hotspot/Performance FAQ.html.

Дисплей

- Качество цветопередачи высокое, разрешение 1024x768 (рекомендуется 1600x1200).

Платформа для интегрированной среды разработки

Eclipse 3.4.

- Необходимая версия платформы Eclipse автоматически устанавливается вместе с Together. Поддерживается сайт обновления Eclipse для установки Together в уже установленную среду Eclipse 3.4.

Java-платформа

Sun J2SE 5. 0 Update 14.

- Необходимая версия Java Runtime Environment устанавливается автоматически вместе с Together.

Веб-браузер

- Любой веб-браузер, поддерживающий фреймы; для платформы Windows рекомендуется Microsoft Internet Explorer версии 6.0 или более поздней.

2.3 Применение Borland Together на различных этапах проекта

ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Применяя средства **UML-моделирования**, бизнес-аналитики, помимо текста требований, который становится потом техническим заданием на разработку продукта, нередко создают набор моделей, позволяющих лучше понять эти требования. Такие модели могут быть весьма разнообразны: среди них как минимум могут быть описания как имеющегося состояния бизнес-процессов, которые предполагается автоматизировать (данные модели называются AS IS — "как есть"), так и состояния бизнес-процессов, в котором они должны эволюционировать после внедрения продукта (модель TO BE — "как должно стать"). Кроме того, модели могут отличаться способами описания бизнес-процессов, составных частей проекта и взаимодействия с ними пользователей.

Из диаграмм, наиболее часто включаемых бизнес-аналитиками в модели, создаваемые на этапе предпроектного обследования, следует отметить в первую очередь **Use Case-диаграммы** (диаграммы сценариев взаимодействия пользователя с продуктом с точки

зрения пользователя; рис. 1); **диаграммы последовательностей**, описывающие порядок передачи сообщений от одних объектов к другим (рис. 2); диаграммы кооперации, описывающие взаимодействие объектов друг с другом, а также диаграммы деятельности, описывающие потоки работ и изменение состояний объектов.

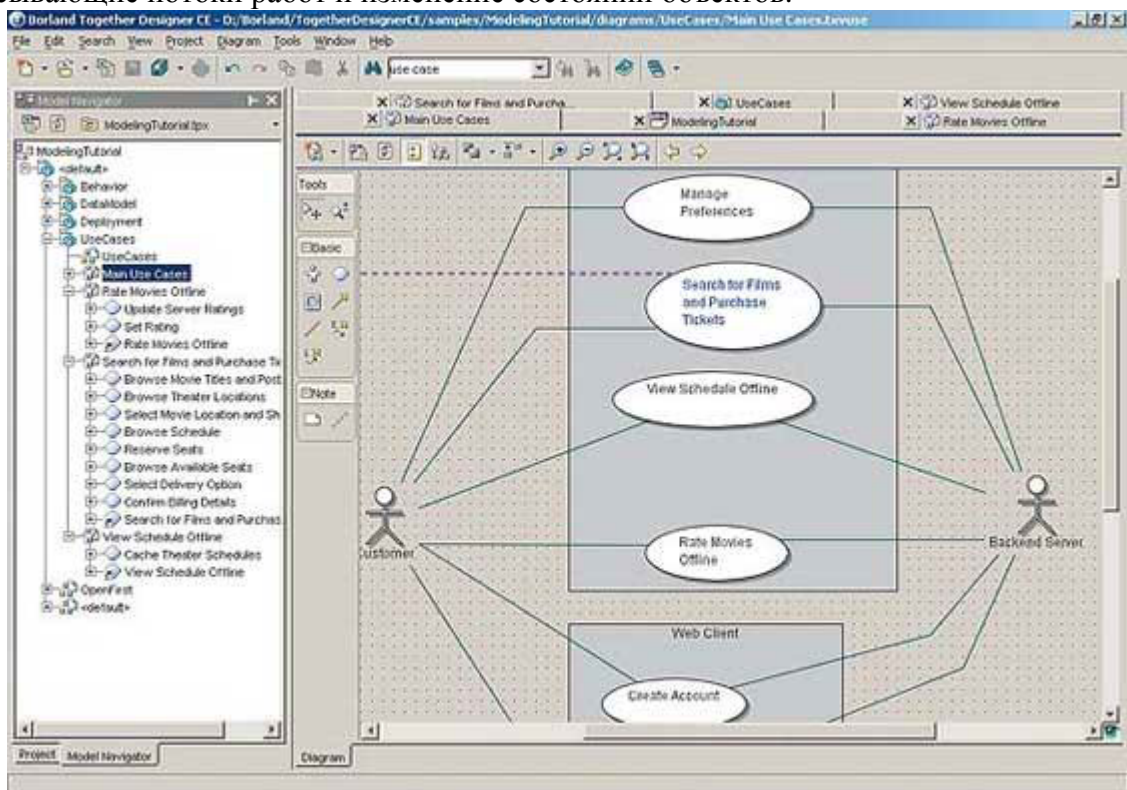


Рис. 1. Use Case-диаграмма в Borland Together

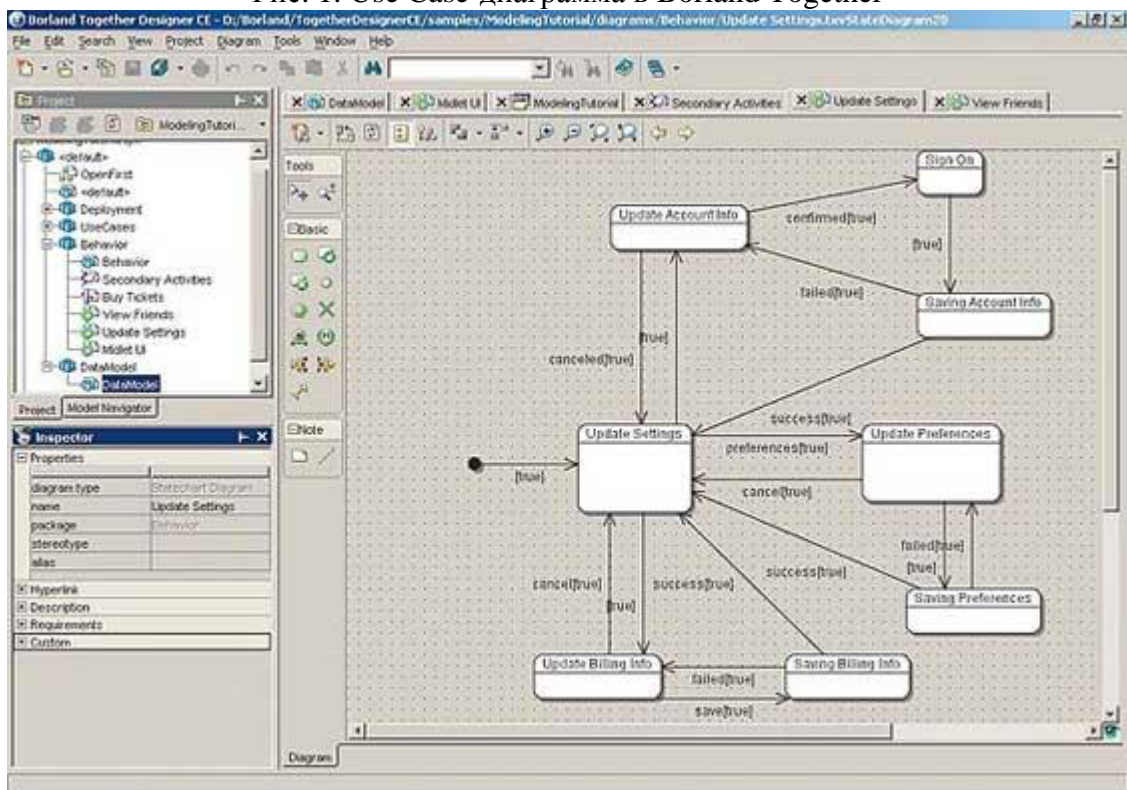


Рис. 2. Диаграмма последовательностей в Borland Together

Модели, созданные на этапе определения требований, нередко упрощают понимание требований и заказчиками и разработчиками за счет более наглядного представления процессов, чем их текстовое описание, и служат для дальнейшего проектирования приложений. Однако стоит заметить, что многие современные инструменты

моделирования, в том числе и Borland Together, отнюдь не являются просто средствами создания иллюстраций — работу над моделями, созданными бизнес-аналитиками, продолжают другие участники проекта, используя модели для генерации кода, для генерации проектной документации, для создания планов тестирования и даже для контроля качества созданного приложения.

ПРОЕКТИРОВАНИЕ

На этапе проектирования принимаются решения относительно архитектуры и составных частей создаваемого решения, а также отбираются технологии, применяемые при его реализации. Обычно на этом этапе к модели, созданной бизнес-аналитиками, добавляются диаграммы компонентов и классов создаваемых приложений и диаграммы развертывания создаваемого решения (рис. 3), а также модифицируются некоторые из уже созданных диаграмм.

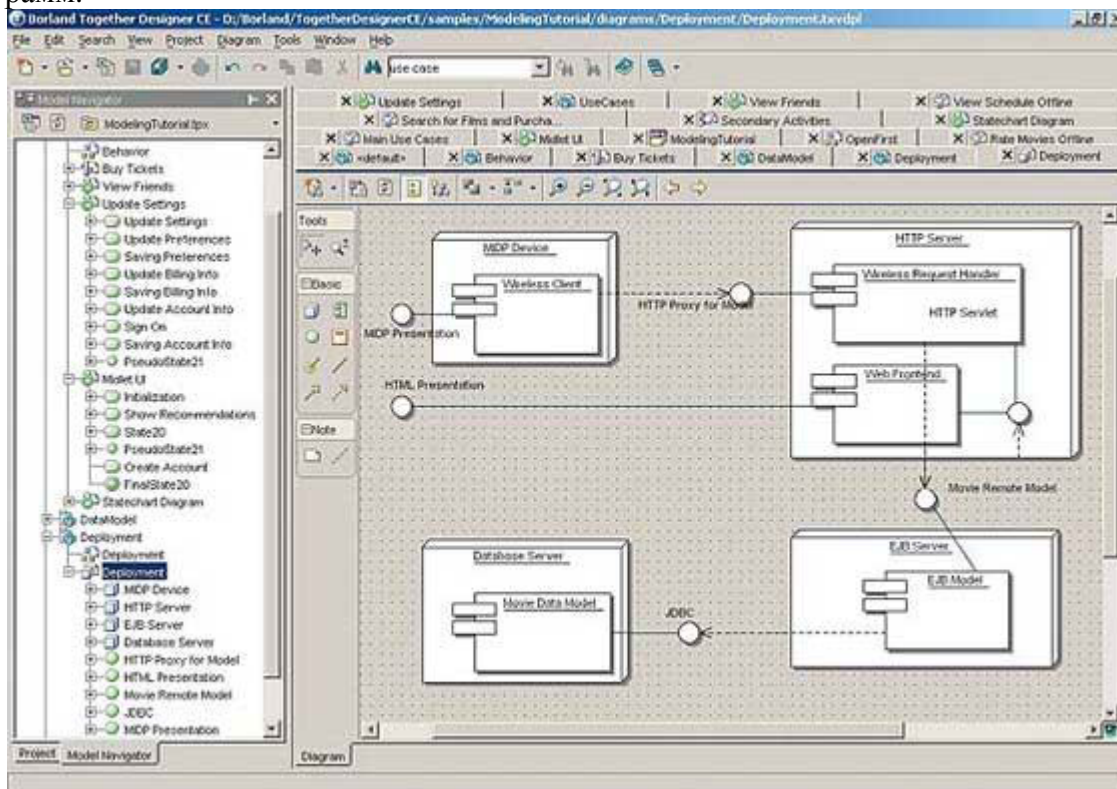


Рис. 3. Диаграмма развертывания

Отметим, что **Together** позволяет применять шаблоны проектирования, что во многом упрощает проектирование и создание часто встречающихся компонентов решений, а также создание компонентов и даже целых проектов с заранее заданными свойствами. На этом же этапе при необходимости может создаваться логическая модель данных, содержащая диаграммы "сущность—связь" (рис. 4), на ее основе генерируется физическая схема данных для конкретной СУБД, выбранной для реализации проекта.

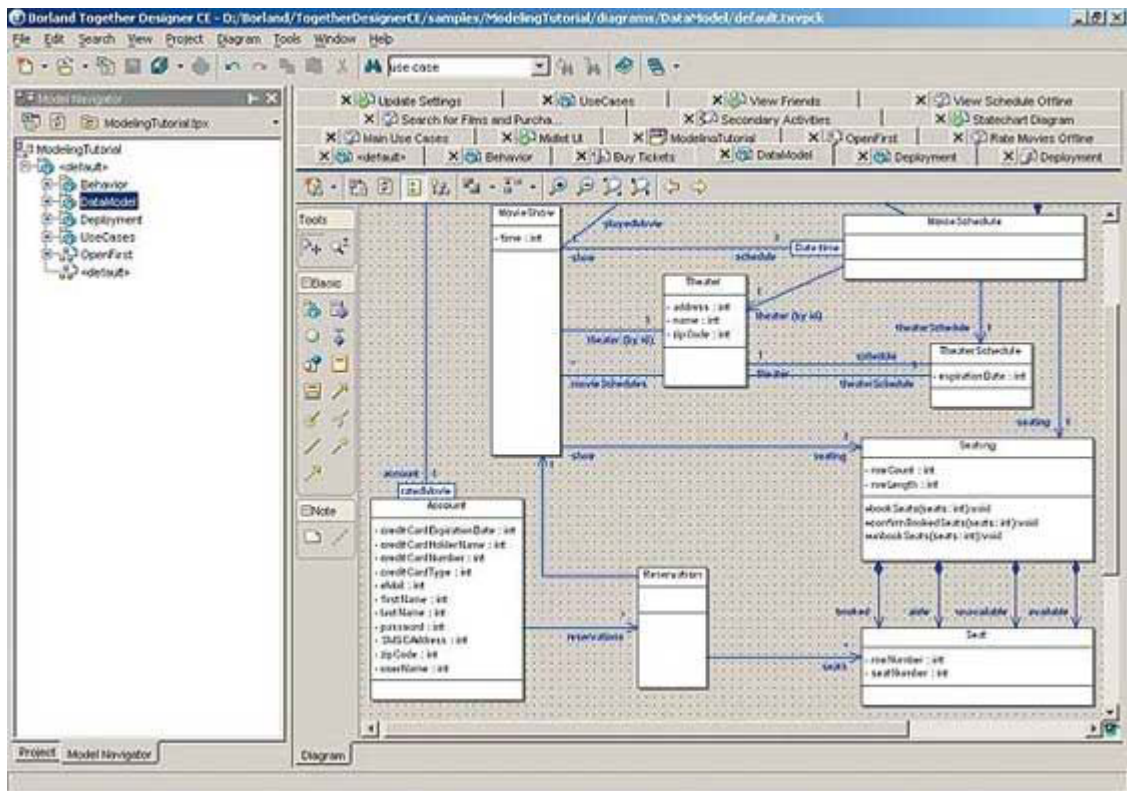


Рис. 4. Диаграмма "сущность—связь"

СОЗДАНИЕ КОДА

На этапе создания клиентского и серверного кода моделирование может применяться весьма активно — практически все современные средства **UML-моделирования** могут осуществлять генерацию кода на различных языках программирования (этот процесс называется **forward engineering**), а некоторые из них могут осуществлять и обратное проектирование (reverse engineering), создавая диаграмму классов на основе готового приложения. Технология **Borland LiveSource** и сами продукты семейства **Borland Together** (в которых эта технология впервые появилась) позволяют не думать о вопросах синхронизации кода и моделей, обеспечивая непосредственную работу с кодом, представленным в виде соответствующих **UML-диаграмм**. Кроме того, прямая работа с кодом позволяет естественным образом интегрировать в среду разработки не только средства **UML-проектирования**, но и инструменты рефакторинга (то есть средства автоматического преобразования кода с целью оптимизации, упрощения "читаемости" при переименовании классов или методов), как это сделано, например, в новой версии **Borland Together Edition for Visual Studio .NET 2.0**

Возможность синхронного изменения кода и модели значительно упрощает многие процессы, сопровождающие собственно разработку. Во-первых, в этом случае не требуется ручного восстановления модели на основе уже созданного кода для создания проектной документации, отражающей реальное положение вещей. Во-вторых, модель уже созданного решения может быть повторно использована в других проектах (возможно, с определенными модификациями) и в этом случае собственно код можно сгенерировать для другого языка программирования, другого средства разработки и даже для другой платформы.

ТЕСТИРОВАНИЕ

Поскольку при тестировании проверяется соответствие продукта требованиям, план тестирования создается именно на их основе, а следовательно, при его создании можно применять модель, созданную на этапе определения требований. Сценарии взаимодействия пользователей с системой выполняются при тестировании пользовательских интерфейсов и при функциональном тестировании.

Помимо тестирования в последнее время немалое внимание стало уделяться оценке качества кода. А поскольку Together обладает средствами разбора кода приложений, то они используются в качестве основы для встроенных в этот продукт средств аудита кода.

СОЗДАНИЕ ДОКУМЕНТАЦИИ

Кроме диаграмм, современные средства моделирования позволяют создавать разнообразные отчеты по моделям и включать их в проектную документацию, и чем аккуратнее велась работа над моделью, тем проще эту документацию создавать. В то же время модель может содержать практически все, что требуется отразить в документации, причем не только в руководстве системного администратора или администратора баз данных, но и в руководстве пользователя. Borland Together позволяет создавать документацию в соответствии с шаблонами и регламентами, настроенными с учетом стандартов страны или отрасли, специфики данной организации, используемой методологии или проекта.

ВНЕДРЕНИЕ И СОПРОВОЖДЕНИЕ

На этапе внедрения продукта специалистами по внедрению обычно применяются диаграммы развертывания, созданные системными аналитиками на этапе проектирования приложения. На этапе же сопровождения продукта соответствующим специалистам может понадобиться практически все, что было произведено при работе над проектом, в том числе код и все созданные модели, особенно если на этом этапе выявляется необходимость какой-либо доработки продукта.

КОЛЛЕКТИВНАЯ РАЗРАБОТКА ПРИЛОЖЕНИЙ

Поскольку модели могут применяться практически всеми участниками проекта и подвергаться постоянным изменениям, довольно серьезной может оказаться проблема управления версиями одной и той же модели. Именно поэтому современные средства моделирования обладают, как правило, механизмами интеграции со средствами управления изменениями и средствами контроля версий, что, в свою очередь, позволяет осуществить корректное применение одной модели коллективом участников проекта. Что касается Borland Together, то этот продукт интегрируется со средством управления изменениями **Borland StarTeam**, а также со средством управления требованиями **Borland Caliber RM**.

ПОДДЕРЖИВАЕМЫЕ ПЛАТФОРМЫ

Borland Together можно применять на различных платформах — Windows (поддерживаются версии Windows 2000, Windows XP Professional, Windows NT 4.0 SP6a), Linux (поддерживаются версии Red Hat Linux 7.2, 7.3 и 8.0, SuSE Linux 8.0 и 8.1, Solaris (версии 8 и 9), Mac OS X (версия 10.2.2).

3 Методические указания по выполнению работы

3.1 Постановка задачи

Изучите материал ЛР и сделайте заключение о достаточности функциональных возможностей данного CASE – средства.

3.2 Анализ функциональных возможностей CASE-средств

Сначала изучите функциональные возможности данного CASE – средства. Затем обоснуйте достаточность или недостаточность набора создаваемых моделей для проектирования ИС.

Контрольные вопросы

1 Поясните назначение основных компонентов CASE – средства.

- 2 Какие виды моделей можно создавать с помощью данного CASE – средства?
- 3 Какова обобщенная структура CASE – средства?

Лабораторная работа 4

Анализ возможностей интегрированного комплекса CASE-средств ARIS

Цель работы:

1. Закрепить основные принципы построения CASE-средства ARIS.
2. Научиться использовать CASE-средство ARIS.

1 Задание на лабораторную работу

- 1.1 Анализ функциональных возможностей CASE-средств а ARIS.
- 1.2 Методика применения CASE-средства ARIS.

2 Краткие теоретические сведения

2.1 Состав CASE-средств

Что такое ARIS

ARIS — это методология и базирующееся на ней семейство программных продуктов, разработанных компанией [IDS Scheer AG](#) (Германия) для структурированного описания, анализа и совершенствования бизнес-процессов предприятия, подготовки к внедрению сложных информационных систем и контроллинга бизнес-процессов. Программные продукты ARIS занимают лидирующие позиции на мировом рынке в классе средств моделирования и анализа бизнес-процессов (источник: Gartner Group). Они используются на всех этапах цикла работ по созданию и развитию бизнеса: при разработке стратегии компании, реорганизации основополагающих бизнес-процессов и организационной структуры, при управлении стоимостью бизнес-процессов и мониторинге их выполнения, при внедрении и поддержке информационных систем класса ERP, CRM, Workflow.

IDS Scheer Aris Platform 7.1.0(2011)

Интегрированный пакет программных продуктов, позволяющих компаниям постоянно совершенствовать свои бизнес-процессы, который охватывает все этапы BPM-проекта: от определения стратегии и проектирования процессов до импорта моделей в ИТ-системы и мониторинга выполнения процессов.

Программное обеспечение ARIS Platform успешно используется для построения сервисно-ориентированной архитектуры (SOA) на предприятиях и применяется в таких областях, как производство товаров народного потребления, продуктов химической, фармацевтической, бумажной, металлургической и текстильной промышленности. Интеграция с различными методами и схемами моделирования (EPC, UML, BPEL, BPMN, ITIL, TOGAF, DoDAF, TEAF/FEAF, ArchiMate и Zachman) позволяет внедрять ARIS в различных бизнес-сферах.

Программное обеспечение ARIS Platform состоит из четырех платформ:

ARIS Strategy Platform – решение для проектирования бизнес-стратегии, внедрения ее в бизнес-процессы и оперативного мониторинга стратегических целей.

ARIS Design Platform – программа распределенного моделирования, оптимизации, публикации бизнес-процессов и управления ИТ-архитектурой.

ARIS Implementation Platform – приложение для подготовки моделей бизнес-процессов к SAP NetWeaver, создания SOA, управления процессом разработки программного обеспечения и менеджмента деловых правил.

ARIS Controlling Platform – платформа контроля бизнес-процессов и внедрения согласованной корпоративной системы управления в рамках всего предприятия.

ARIS Strategy Platform позволяет создавать системы сбалансированных показателей и оптимизировать бизнес-процессы в соответствии с ними. Модули, относящиеся к платформе:

- [ARIS BSC](#) – разработка системы сбалансированных показателей (ССП)
- [ARIS Business Optimizer](#) – расчет показателей процесса (КПП)
- [ARIS Business Simulator](#) – имитационный анализ процессов (расчет времени, стоимости процессов)

ARIS Design Platform позволяет выявлять организационные, структурные и технологические недостатки и определять возможности оптимизации. Модули, относящиеся к платформе:

- [ARIS Business Architect](#) – профессиональное моделирование и управление бизнес-процессами
- [ARIS Business Designer](#) – клиент-серверное моделирование бизнес-процессов
- [ARIS Business Publisher](#) – создание корпоративных WEB-порталов по бизнес-процессам
- [ARIS Process Governance](#) – управление изменениями бизнес-процессов
- [ARIS IT Inventory](#) – инвентаризация ИТ-приложений
- [ARIS IT Architect](#) – моделирование ИТ-архитектуры предприятия
- [ARIS Business Server](#) – централизованное хранение и управление моделями
- [ARIS Express](#) – бесплатный инструмент для моделирования бизнес-процессов

ARIS Implementation Platform позволяет реализовывать бизнес-процессы в ИТ-среде. Модули, относящиеся к платформе:

- [ARIS Business Architect for SAP](#) – интеграция моделей бизнес-процессов с SAP
- [ARIS BI Modeler](#) – редокументирование бизнес-процессов
- [ARIS SOA Architect](#) – моделирование бизнес-процессов в SOA-проектах
- [ARIS Business Rules Designer](#) – управление правилами в бизнес-процессах
- [ARIS UML Designer](#) – моделирование бизнес-процессов при разработке бизнес-приложений.

ARIS Controlling Platform используется для поиска возможностей совершенствования путем оценки и визуализации выполненных процессов, импортированных из ИТ-систем. Модули, относящиеся к платформе:

- [ARIS Risk & Compliance Manager](#) – система контроля за соответствиями на основе технологии Workflow
- [ARIS Process Performance Manager](#) – мониторинг бизнес-процессов в реальном времени
- [ARIS Process Event Monitor](#) – контролинг бизнес-процессов
- [ARIS MashZone](#) – создание управленческих панелей

2.2 Функциональные возможности CASE-средств

В рейтинге Gartner Group система ARIS занимает лидирующее положение на рынке средств моделирования и анализа деловых процессов.

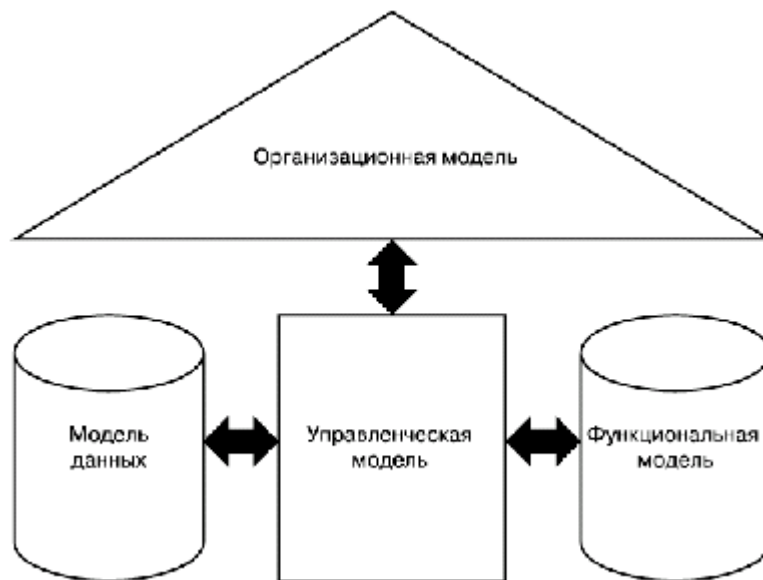


Рисунок 2 - Взаимосвязь типов моделей, используемых ARIS архитектура ARIS

В рамках каждого из перечисленных типов создаются модели разных видов, отражающие соответствующие стороны исследуемой системы. ARIS поддерживает большое количество методов моделирования, используемых для построения этих моделей. Среди них такие известные как диаграммы Чена, Unified Modeling Language (UML), Object Modeling Technique (OMT) и т.п. Последняя версия ARIS поддерживает более 83 методов моделирования.

Другой особенностью методологии ARIS, обеспечивающей целостность разрабатываемой системы, является использование различных уровней описания, что поддерживает теорию жизненного цикла системы, существующего в сфере информационных технологий. В ARIS Toolset используется трехфазовая модель жизненного цикла, т.е. каждый из перечисленных аспектов имеет три уровня представления:

1. **Уровень определения требований.** На данном уровне разрабатываются модели, описывающие то, что должна делать система - как она организована, какие деловые процессы в ней присутствуют, какие данные при этом используются.
2. **Уровень проектной спецификации.** Этот уровень соответствует концепции информационной системы, определяющей основные пути реализации предъявленных на втором этапе требований.
3. **Уровень описания реализации.** На данном этапе жизненного цикла создания информационных систем происходит преобразование спецификации в физическое описание конкретных программных и технических средств. Это заключительный этап проектирования систем, за которым следует этап физической реализации (программирования). Уровень описания реализации порождает документы, на основе которых можно обеспечить процесс разработки программных модулей (или подбора готовых программных компонент, отвечающих поставленным требованиям), а также выбора и организации технических средств реализации системы.

Графически такое разделение по уровням в ARIS выглядит следующим образом:



Рисунок 3- Уровни представления моделей в ARIS

3 Методические указания по выполнению работы

3.1 Постановка задачи

Изучите материал ЛР и сделайте заключение о достаточности функциональных возможностей данного CASE – средства.

3.2 Анализ функциональных возможностей CASE-средств

Сначала изучите функциональные возможности данного CASE – средства. Затем обоснуйте достаточность или недостаточность набора создаваемых моделей для проектирования ИС.

3.3 Методика применения CASE-средства ARIS.

ИСПОЛЬЗОВАНИЕ ARIS

Итак, ARIS представляет собой инструмент, используемый в процессе консалтинга с целью реорганизации предприятий. Какова же технология использования такой системы? Вообще говоря, из-за широкого спектра возможностей ARIS такая технология может значительно варьироваться в зависимости от целей проекта, реализуемого с помощью данной системы. Тем не менее, в общем случае можно выделить некоторую последовательность этапов такой технологии. Эта последовательность может включать следующие этапы:

- сбор информации об объекте;
- построение модели объекта "как есть";
- анализ построенных моделей;
- построение моделей "как должно быть";
- проектирование информационной системы (в случае необходимости).

Сбор информации является первым этапом реализации проекта по реорганизации предприятия. Полученная в результате исследования объекта информация требует формализации и структуризации. На этом этапе и вступает система ARIS. С помощью данного программного продукта строятся модели объекта, отражающие его жизнедеятельность с разных сторон. Это может быть организационная структура предприятия, дерево его целей, информационные модели, отражающие структуру

информации, используемой на предприятии и т.п. Конкретный набор методов, используемых для построения таких моделей, формируется в зависимости от целей проекта из числа тех восьмидесяти трех методов, которые поддерживает система ARIS.

При этом также определяется, какие из четырех аспектов архитектуры ARIS (организационная структура, функциональная составляющая, информационная составляющая и процессная составляющая) найдут свое отображение в итоговом наборе моделей. Для всестороннего описания системы необходимо построение моделей, отображающих все перечисленные аспекты, но для некоторых проектов полный набор может быть избыточен и потребуются исследование одного-двух аспектов. Например, организационного и процессного.

Модели в ARIS представляют собой графические схемы, отображающие соответствующие аспекты системы.

Элементами таких схем являются объекты, поддерживаемые ARIS. В качестве примеров объектов можно привести такие как "Функция", "Событие", "Структурное подразделение", "Документ" и т.п. Между объектами устанавливаются разнообразные связи. Так, между объектами "Функция" и "Структурное подразделение" могут быть установлены связи следующих видов:

- выполняет;
- принимает решение;
- участвует в выполнении;
- должен быть проинформирован о результатах;
- консультирует исполнителей;
- принимает результаты и т.п.

Каждому объекту соответствует установленный для объекта данного типа набор атрибутов, которые позволяют ввести дополнительную информацию о конкретном объекте. На следующих этапах введенные значения атрибутов могут использоваться при имитационном моделировании или для проведения стоимостного анализа.

Таким образом, по результатам выполнения этого этапа возникает набор взаимосвязанных моделей, представляющих собой исходный материал для дальнейшего анализа. Пример такой модели приведен на рисунке.

Как уже отмечалось, в рамках методологии ARIS поддерживается стоимостной и имитационный динамический анализ. По результатам анализа проектируется новая система, которая должна возникнуть по результатам реорганизации действующей. На этом этапе ARIS используется для построения моделей проектируемой системы, что позволяет формализовать принимаемые в рамках реорганизации решения и облегчить процесс их согласования.

ARIS может использоваться не только как средство реорганизации предприятий с точки зрения преобразования деловых процессов, которые используются на данном предприятии. Важной составляющей функциональных возможностей ARIS является сопровождение процесса проектирования и внедрения автоматизированной информационной системы.

Как отмечалось выше, ARIS поддерживает три основные фазы жизненного цикла информационной системы - определение требований, разработка проектной спецификации, описание реализации. Если учитывать отечественные разработки по этому поводу, то можно отметить, что данные фазы соответствуют принятым в отечественной практике, согласно сборнику ГОСТов "Единая система программной документации", стадиям разработки программного продукта - разработка технического задания на систему, эскизный проект, технический проект.

Таким образом, методология ARIS позволяет осуществить комплексное исследование системы, создать ее формализованное описание, провести анализ полученных моделей, спроектировать информационную систему и описать ее реализацию. В этой методологии наиболее полно воплощаются базовые принципы системного подхода, требующего полного и всестороннего рассмотрения исследуемой системы.

Можно с уверенностью сказать, что на сегодняшний день ARIS является одним из наиболее перспективных инструментов проектирования и анализа систем.

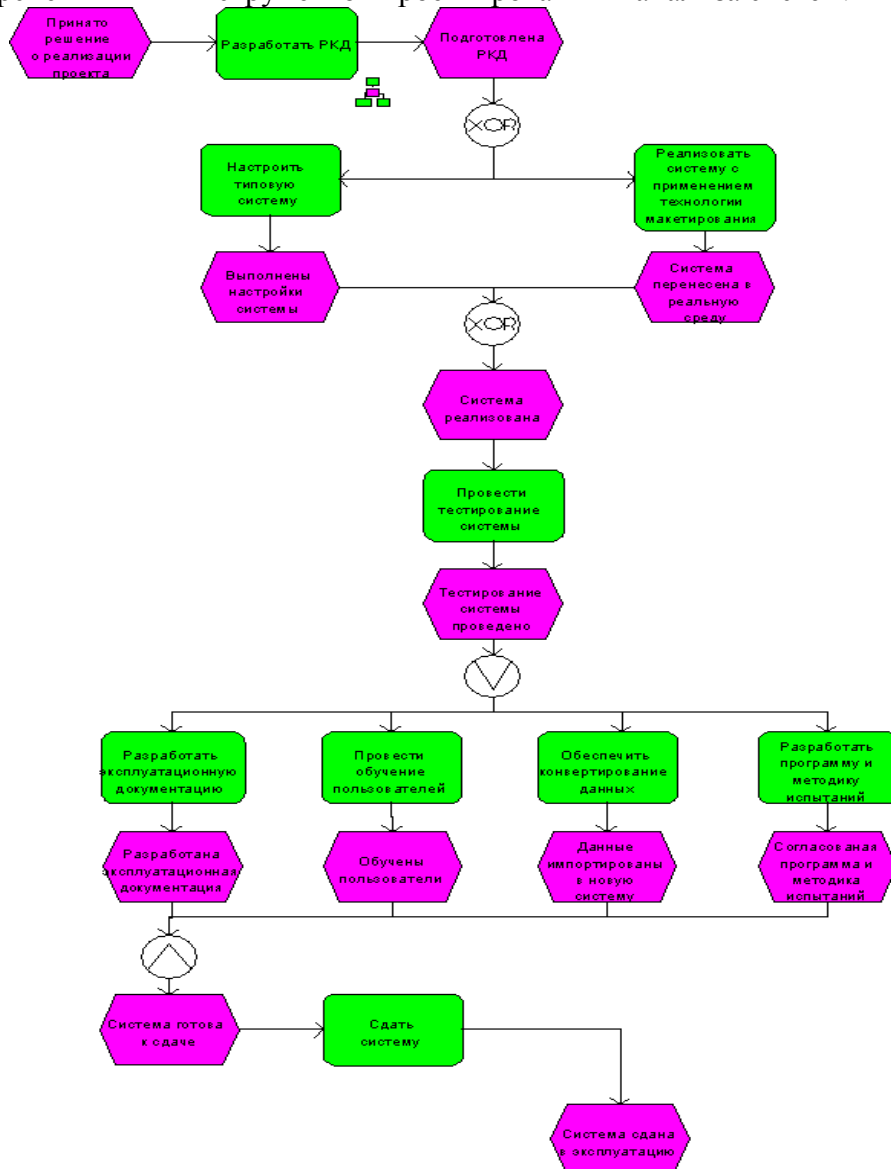


Рисунок 4- Модель процесса проектирования и разработки ИС

Контрольные вопросы

- 1 Поясните назначение основных модулей ARIS
- 2 Какие виды моделей можно построить с помощью ARIS?
- 3 Каковы уровни представления моделей в ARIS?

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

6.1.1. Основная литература			
Авторы,	Заглавие	Издательство, год	Адрес
Авлукова, Ю. Ф.	Основы автоматизированного проектирования: учебное пособие	Минск: Вышэйшая школа, 2013	http://www.iprbookshop.ru/2407
Бурков А. В.	Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008	Москва: Интернет - Университет Информационных	http://www.iprbookshop.ru/52166.html
6.1.2. Дополнительная литература			
Авторы,	Заглавие	Издательство, год	Адрес
Жданов, С. А., Соболева, М. Л., Алфимова,	Информационные системы: учебник для студентов учреждений высшего образования	Москва: Прометей, 2015	http://www.iprbookshop.ru/5813
Краюткина, Е. В.	Методы и средства проектирования информационных систем и технологий: учебное пособие	Ставрополь: Северо-Кавказский федеральный	http://www.iprbookshop.ru/62959.html
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"			
Задания на лабораторные работы			
Антонов, В. Ф. Методы и средства проектирования информационных систем [Электронный ресурс] : учебное пособие / В. Ф. Антонов, А. А. Москвитин. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016.			
Вичугова, А. А. Инструментальные средства информационных систем [Электронный ресурс] : учебное пособие / А. А. Вичугова. — Электрон. текстовые данные. — Томск : Томский государственный университет, 2015. — 126 с. — 078-5-4297-0574-1. — Режим			



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Модели и методы поддержки принятия решений»
для студентов направления подготовки
09.04.02 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Модели и методы поддержки принятия решений» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1. Исследование с помощью системы "Выбор"

Лабораторная работа 2. Исследование с помощью системы "Выбор"

Лабораторная работа 3. Исследование с помощью системы "Выбор"

Лабораторная работа 4. Исследование с помощью системы "Выбор"

Лабораторная работа 5. Исследование с помощью системы "Выбор"

Лабораторная работа 6. Исследование с помощью системы "Выбор"

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины - развить системное мышление у обучающихся путем детального анализа подходов к математическому моделированию и сравнительного анализа разных типов моделей. Ознакомить обучающихся с математическими свойствами методов и моделей оптимизации, которые могут использоваться при анализе и решении широкого спектра задач. Выработать у обучающихся навыки проведения численных исследований математических моделей и анализа результатов вычислений. Научить выбирать наиболее перспективное управляющее решение.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Изучив данный курс, студент должен:

Знать:

основы моделирования управленческих решений;

математические модели и информационные технологии процесса поддержки принятия решений;

многокритериальные методы поддержки принятия решений;

основные технологии информационной поддержки процесса поддержки принятия решений;

классификацию систем поддержки принятия решений и особенности используемых инструментальных средств;

современные методы и средства поддержки принятия решений в различных интеллектуальных системах,

принципы их рационального выбора в зависимости от особенностей процесса поддержки принятия решений.

Уметь:

осуществлять постановку конкретных задач поддержки принятия решений, выбирать адекватные математические и инструментальные средства их решения;

решать задачи, связанные с различными этапами подготовки и принятия решений в инструментальных системах

Владеть:

навыками формулирования требований к методам и моделям поддержки принятия решений;

навыками разработки отдельных их элементов;

навыками практического использования моделей и методов поддержки принятия решений;

навыками аналитического обоснования вариантов решений с использованием систем поддержки принятия решений.

Реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа 1. Исследование с помощью системы "Выбор"

Цель занятия:

1. Понятие СППР. Эволюция информационных технологий и информационных систем. Усвоить основные теоретико-информационные понятия учебной дисциплины изучить этапы развития информационной технологии и информационных систем.

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения

1. Дайте определение понятию конкурентные гонки?
2. Что такое информация?
3. Определите содержание СППР?
4. Чем отличаются данные от информации?
5. Основные характеристики данных?

Ход работы

1. Информация и данные.
2. Развитие информационных технологий.
3. Перспективные средства и направления развития информационных систем.
4. Основные понятия систем поддержки принятия решений.

Задача 1. Привести по три определения каждого понятия: “информация”, “данные”, “система поддержки принятия решений” (СППР).

Задача 2. Выделить критерии отбора альтернативных вариантов, которые, по вашему мнению, должны входить в состав СПР выбранной тематики. (например инвестиционные проекты: прибыль, срок окупаемости, и прочее). Также нужно выделить главные и второстепенные критерии. обосновать свой выбор. Тема определяется в соответствии с номером студента в академическом журнале (см темы разработки сппр).

Задача 3. Привести несколько существенных преимуществ применения СППР в выбранной области. Ответ обоснуйте.

Темы разработки сппр:

1. Создание проекта выбора ПК
2. Создание проекта выбора ТВ
3. Создание проекта выбора монитора
4. Создание проекта выбора микроволновой печи
5. Создание проекта выбора автомобиля
6. Создание проекта выбора магнитолы
7. Создание проекта выбора принтера
8. Создание проекта выбора сканера
9. Создание проекта выбора плоттера
10. Создание проекта выбора материнской платы
11. Создание проекта выбора процессора
12. Создание проекта выбора модема
13. Создание проекта выбора мобильного телефона
14. Создание проекта выбора программного обеспечения (по)
15. Создание проекта выбора винчестера (HЖМД)
16. Создание проекта выбора стационарного телефона
17. Создание проекта выбора DVD-проигрывателя
18. Создание проекта выбора интернет-провайдера
19. Создание проекта выбора видеокарты
20. Создание проекта выбора факсимильного аппарата

Лабораторная работа 2. Исследование с помощью системы "Выбор"

Тема: работа с СППР "Выбор". Основные функции, приемы и возможности

Цель занятия:

1. Ознакомиться с основными командами и получить базовые навыки при работе с СППР "Выбор"

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения

1. Назовите основные возможности СППР "Выбор";

2. Перечислите основные команды, которые потребуются для создания нового работоспособного проекта.
3. Перечислите основные возможности окна разработки проектов.
4. Перечислите основные преимущества и недостатки СППР “Выбор”.

Ход работы:

1. Ознакомиться с основными принципами метода анализа иерархий.
2. Изучить основные аспекты интерфейса СППР “Выбор”.
3. Ознакомиться с интерфейсом и основными командами для пользования инструментами доступными в СППР “Выбор”.
4. Познакомиться с основными командами СППР “Выбор”.
5. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
6. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

Основные аспекты интерфейса СППР "Выбор".

Система поддержки принятия решений (СППР) “Выбор”- аналитическая система, основанная на методе анализа иерархий (МАИ), является простым и удобным средством, которое поможет структурировать проблему, построить набор альтернатив, выделить факторы, характеризующие их, задать значимость этих факторов, оценить альтернативы по каждому из факторов, найти неточности и противоречия в суждениях лица принимающего решение (ЛПР эксперта), проранжировать альтернативы, провести анализ решения и обосновать полученные результаты. Система опирается на математически обоснованный метод анализа иерархий Томаса Саати.

Клиентское применение СППР “Выбор” обладает интуитивным пользовательским интерфейсом. Главное окно-это инструмент работы над проектом, позволяющий просматривать и редактировать выбранную иерархию проекта.

Основные компоненты интерфейса СППР " Выбор”

Панели инструментов.

С помощью панелей инструментов осуществляется быстрый доступ к основным инструментам приложения сосредоточенным в главном меню. Каждому пункту меню соответствует кнопка быстрого запуска на панели инструментов.

Подсказки.

Практически все элементы окон снабжены подсказками, которые появляются при небольшой задержке курсора мыши на необходимом элементе. Более подробная информация обо всех инструментах приложения содержится в помощи, сосредоточенной в главном меню Помощь.

Контекстно-зависимая помощь

СППР “Выбор” обладает мощной справочной системой по каждому инструменту приложения. Более 300 пунктов помощи помогут вам быстрее разобраться как использовать тот или иной элемент приложения. Помощь также предоставляет поиск информации по ключевым словам и фразам.

Помощь по диалоговым окнам

Все диалоговые окна снабжены справочной информацией. Нажмите клавишу F1 для вызова справки по текущему окну. Панели инструментов полностью копируют элементы главного меню. Каждому элементу главного меню соответствует панель инструментов:

* Файл-открытие, сохранение, закрытие проектов, создание новых, настройка приложения, принтера для печати, а также закрытие приложения.

• Проект - работа с иерархиями, произведение расчетов, получение отчетов, редактирование свойств проекта.

* Сеть-отсылка выбранным экспертам текстовых сообщений, проектов, открытие пришедших сообщений, просмотр списка сетевых событий.

* Помощь-вызов справки и окна информации о приложении.

В главном окне приложения для быстрого вызова некоторых инструментов можно использовать следующие горячие клавиши:

Файл

Ctrl + n-новый
F3-Открыть
F2-Сохранить
Ctrl + f2-Сохранить как
F10-Закреть
F4-настройка принтера
Ctrl+o - Опции

Проект

Ctrl + i - информация о готовности
Ctrl + c-расчет
Ctrl + alt + c - сетевой расчет
Ctrl + r-результаты расчетов
Ctrl + p-просмотр отчета
Ctrl + t-изменить типа
Ctrl+v - Режим просмотра
Shift+Ctrl + P - Свойства

Сеть

F5-текстовое сообщение
F6-послать проект
F10-выбрать экспертов
F9-Эксперты
F11 - Окно сетевых сообщений
F12-окно с пакетами

Помощь

F1-Информация
Ctrl + f1-о программе

В окнах со списками, если фокус ввода находится на списке, то становятся доступными следующие горячие клавиши:

- Insert или "+" - добавление записи.
- * Delete или "-" - удаление текущей записи.

Практическое задание.

Создать проект выбора автомобиля имея такую информацию:

Марки автомобилей 5-10 шт.

Критерии оценки:

1. Мощность двигателя (мощный)
2. Цена (Высокая)
3. Качество (Высокое)
4. Ведомость марки (Известная)
5. Комфортность (Комфортная)
6. Стильность (Стильная)

Создание проекта

1. Командой “Файл” – “Новый...” – “Простой проект” создать новый проект.
2. Щелкнуть на прямоугольнике по центру окна правой кнопкой мыши, вызвав при этом контекстное меню и в нем выбрать команду “Свойства проекта”.
3. Выбрать вкладку "Уровни".

4. Щелкнуть на знаке “+” 3 раза.
5. В списке появится 3 уровне (рис. 1).

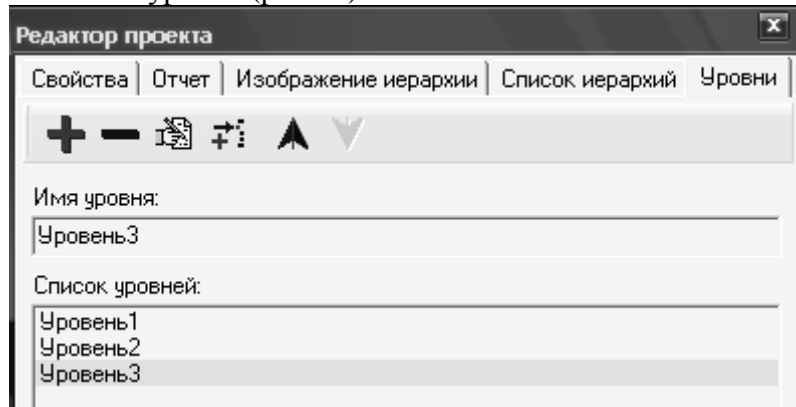


Рис. 1. Окно “Редактор проекта”.

6. Нажимая на каждый из уровней нужно изменить имя каждого из них. Уровень1 – Покупка автомобиля, Уровень2 – Критерии, Уровень3 – Марки авто.

7. Далее нужно нажать на “Покупка автомобиля”, которое находится в списке дважды левой кнопкой мыши.

8. Появится окно “Редактор уровня3”.

9. Нужно щелкнуть на вкладке “Узлы3, и нажимая на красный знак “+” добавить узел, и изменить его имя на “Цель”.

10. Затем нужно нажать ОК и сделать так же с уровнями 2 и 3. Во второй уровень нужно добавить 6 узлов и назвать их критериям условия, в третий уровень нужно добавить минимум 5 узлов и вписать любые марки машин так, чтобы каждой машине соответствовали по крайней мере 2 критерии.

Замечание !!! Для облегчения понимания программы рекомендуем писать только положительные качества предметов цели, то есть если идет речь о выборе машин и о критериях “цена” нужно в критерии добавить узел “высокая цена”. Если авто с низкой ценой между узлом марки авто и узлом “высокая цена” просто не делать связь.

11. Теперь нужно установить связь между каждым узлом. Для этого нужно вызвать окно “Редактор уровня” используя шаги 7 и 8. Окно СППР “Выбор” к установлению связей показаны на рис. 2:

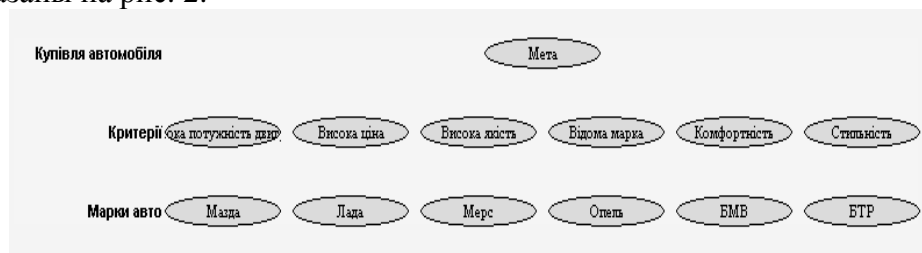


Рис. 2. Окно СППР “Выбор” к установлению связей.

12. Для того чтобы установить связи каждого из узлов нужно дважды щелкнуть на имени узла и в окне “Редактор узла” выбрать вкладку “Связи”.

13. Входных связей первого уровня не будет поэтому нужно выбрать вкладку “Исходящие” и выбрать нажав на стрелку возле выпадающего списка, выбрав в нем критерии.

14. В списке ниже появятся критерии, требуется выбрать все.

15. Так же нужно сделать с уровнем Критерии.

16. Установив необходимые связи получим готовый для расчетов проект (рис. 3.).

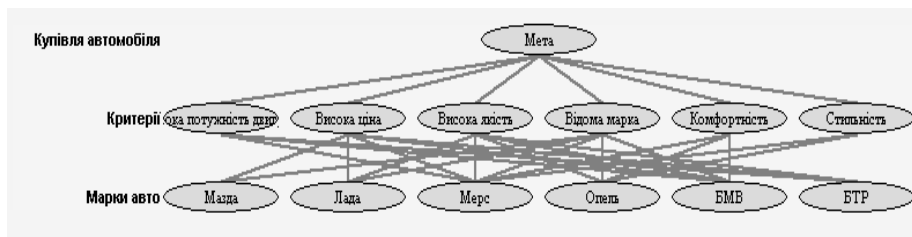


Рис. 3. Окно СППР “Выбор” после установления связей.

Для проверки правильности проделанной работы нужно запустить процесс расчета командой “Проект” – “Расчет”, если все сделано верно, начнется процесс подсчета. В противном случае нужно исправить ошибки. Самая распространенная ошибка-недостаточность связей между узлами, что легко исправить добавив их или заменив другими.

Далее нужно сохранить проект командой “Файл” – “Сохранить”.

Варианты заданий:

1. Создать проект выбора мобильных телефонов
2. Создать проект выбора телевизоров.
3. Создать проект выбора холодильников.
4. Создать проект выбора процессоров.
5. Создать проект выбора ОЗУ.
6. Создать проект выбора Винчестеров.
7. Создать проект выбора колонок.
8. Создать проект выбора магнитол.
9. Создать проект выбора проводов.
10. Создать проект выбора мониторов.
11. Создать проект выбора мешков.
12. Создать проект выбора клавиатур.
13. Создать проект выбора материнских плат.
14. Создать проект выбора принтеров.
15. Создать проект выбора модемов.
16. Создать проект выбора ксероксов.
17. Создать проект выбора стационарных телефонов.
18. Создать проект выбора охлаждающих систем (имеется в виду кулеры или водяное охлаждение).
19. Создать проект выбора тюнеров.
20. Создать проект выбора бесперебойников.

Лабораторная работа 3. Исследование с помощью системы "Выбор"

Тема: работа с СППР "Выбор". Расчеты, представление информации, выводы.

Цель занятия:

1. Ознакомиться с основными командами и получить базовые навыки при работе с СППР “Выбор”.

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения:

1. Охарактеризовать принципы работы матрицы парных сравнений.
2. Охарактеризовать окно “Результат вычислений”.
3. Перечислите команды, которые нужно выполнить для изменения вида диаграммы.
4. Для чего нужен индекс согласованности.
5. Для чего нужны веса у исследуемых альтернатив?

Ход работы:

1. Изучить основные аспекты вычислений в СППР “Выбор”.
2. Ознакомиться с интерфейсом и основными командами для проведения вычислений в СППР “Выбор”.
3. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
4. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

задание

Провести нужные расчеты и оформить выводы с предварительно сделанным проектом выбора машины.

Ход выполнения работы

1. Открыть предыдущий проект командой "Файл" - "Открыть".
2. Командой “Проект” – “Расчет”, вызвать окно, изображенное на рис. 1:

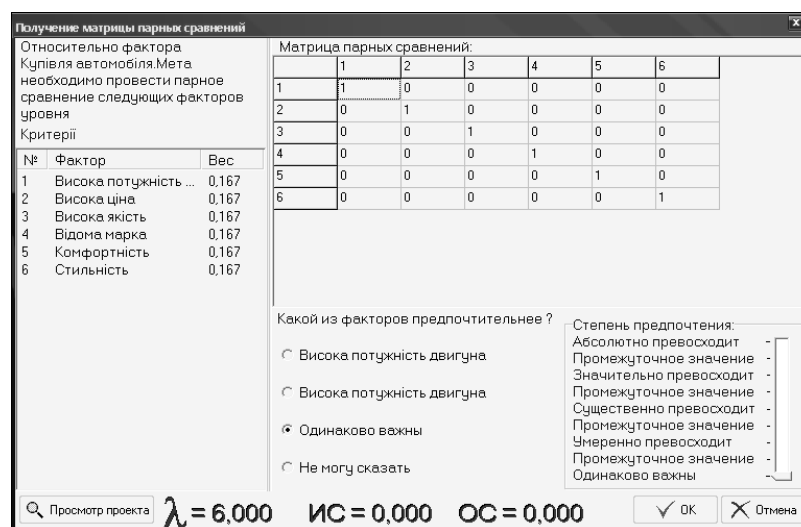


Рис. 1. Окно «Получение матрицы парных сравнений»

3. Данное окно представляет собой нечто вроде тестирующей программы оператором которого является матрица, которую нужно заполнить значениями. Для этого нужно нажимать на каждой ячейке под главной диагональю и отвечать на вопросы. когда

каждое поле будет заполнено нужно нажать ОК программа обработает информацию и выведет следующее окно (рис. 2).

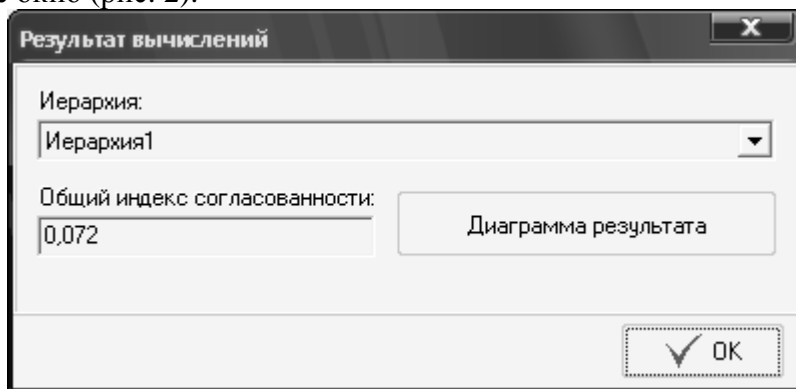


Рис. 2. Окно “Результат вычислений (анализ “Проблема выбора ”)”.

4. Индекс согласованности показывает на то что данные которые Вы ввели не противоречивыми.

5. Далее нужно вывести диаграмму результата нажав на аналогичную клавишу в окне “Результат Вычислений (рис. 3):

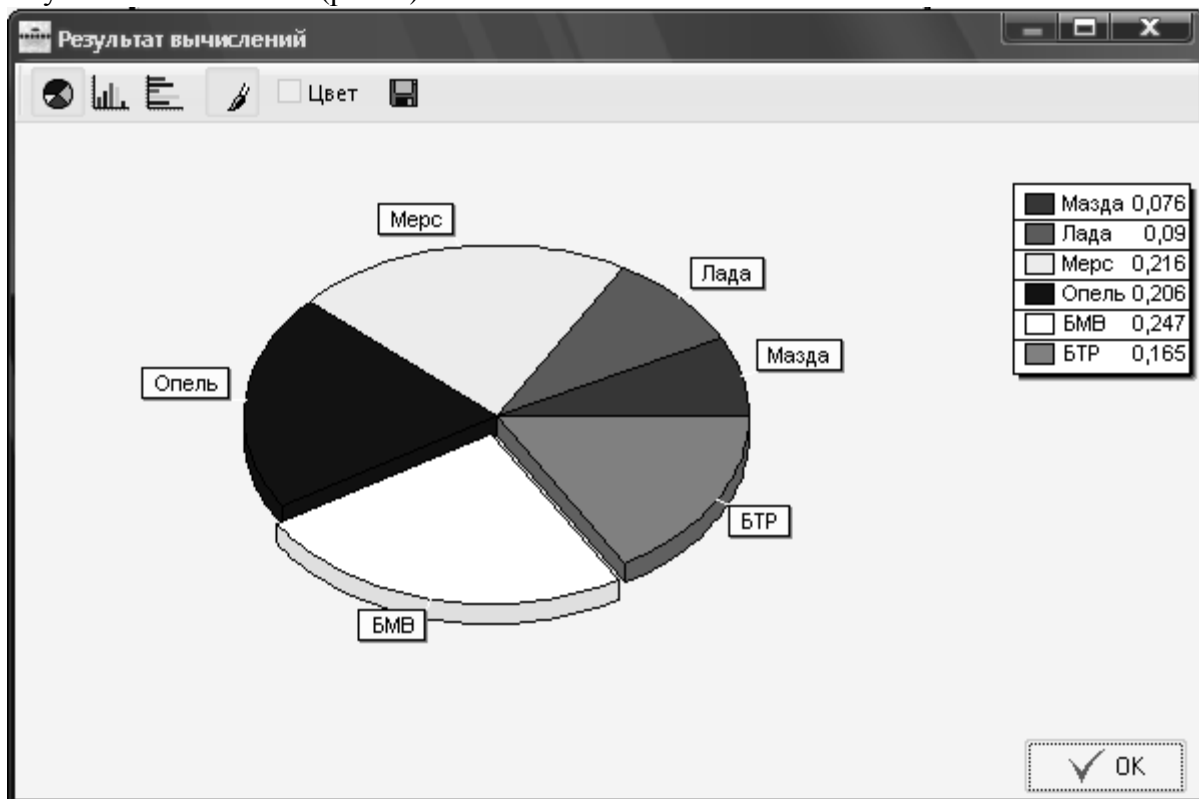


Рис. 3. Окно “Результат Вычислений”

6. В данном окне можно выбрать вид диаграммы, цвет, можно также сохранить ее. Нужно сохранить проект. Диаграмма и расчеты сохраняются вместе с проектом.

Лабораторная работа 4. Исследование с помощью системы "Выбор"

Тема: работа с СППР"Выбор". Создание проекта типа “Стоимость-эффективность”

Цель занятия:

- 1.Изучение возможностей проекта типа “Стоимость-эффективность” в программе «Выбор» как элемента моделирования СППР
2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;
УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения:

1. Чем отличаются проекты “Проблема выбора” и “Стоимость – эффективность”?
2. Для чего нужны уровни критериев и альтернатив?
3. Охарактеризуйте структуру иерархии в СППР “Выбор”.
4. Преимущества и недостатки СППР “Выбор”.

Ход работы

1. Проработать теоретический материал, рекомендации по работе с программой.
2. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
3. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

Постановка задачи и суть проекта типа “Стоимость-эффективность”.

Проекта типа “Стоимость-эффективность”.- это проект, состоящий из двух иерархий, иерархии выгод и иерархии затрат, которые впоследствии необходимо будет между собой ранжировать (сравнивать).

Например, рассматривается задача принятия решения по множеству планов или каких-либо проектов. Классический подход основан на оценке каждого проекта с точки зрения затрат (то есть, сколько необходимо сделать инвестиций для реализации данного проекта) и с точки зрения доходов, которые можно получить при их реализации. Сравнение альтернативных проектов сводится к сравнению объемов доходов из расчета на единицу ресурса (т. е. расходов). Этот метод известен как анализ “Стоимость-эффективность”.

Для примера можно предложить задачу по решению целесообразности выбора определенного проекта. Решение задачи ранжирования проектов а, в, С при традиционном подходе может быть сведено в табл. 1.

Таблица 1. Классический подход к задаче “Стоимость-эффективность”

Проект	Расходы	Доходы	Доходы/ расходы	Ранжирования
А	500	1000	2	3
В	250	750	3	1
С	600	1300	2,1	2

При решении этой задачи возникают следующие особенности:

□ отношение доходов к затратам, оценивается в стоимостном выражении, по сути не является объективной мерой качества проекта: неясно, как, например, оценивать в

деньгах выгоды и затраты неосязаемых количественно показателей (то есть проблема измерения качественных факторов);

□ известно также, что доходы и расходы распределяются по многим сферам - социальным, экономическим, политическим, управленческим и их взаимосвязь влияет на оценку альтернатив.

Применение СППР позволяет снять эти проблемы. В этом случае нужно построить две иерархии: одну для затрат, другую для выгод с одними и теми же альтернативами на нижнем уровне. Таким образом, получают два вектора приоритетов - доходов и расходов. Затем вычисляют отношения доходов к расходам для каждой альтернативы. Наибольшее значение из этих отношений и определяет лучший проект.

В целом решение задач типа "Стоимость-эффективность" происходит в следующем порядке:

- а) правильная формулировка цели;
- б) построение иерархии выгод (все те же действия, что и в п. 1, но отталкиваться лишь от выгод, не обращает внимания на затраты);
- в) построение иерархии издержек (все те же действия, что и в п. 1, но отталкиваться лишь от издержек, не обращающих внимания на выгоды).

Пример решения.

Выбор темы разработки СППР происходит согласно номера студенте в академическом журнале.

Создать СППР для выбора инвестиционного проекта имея такую информацию:

Количество проектов-4.

Критерии оценки (позитивные или выгоды):

7. Надежность;
8. Быстрая окупаемость;
9. Доходность.

Критерии оценки (отрицательные, или расходы):

1. Сложность контроля;
2. Жаль н/с-щу;
3. Удаленность.

Создание проекта

1. Командой "Файл" – "Новый..." – "Простой проект" создать новый проект.
2. Щелкнуть на прямоугольнике по центру окна правой кнопкой мыши, вызвав при этом контекстное меню и в нем выбрать команду "Свойства проекта".
3. Выбрать вкладку "Уровни".
4. Щелкнуть на знаке "+" 3 раза.
5. В списке появится 3 уровне.
6. Нажимая на каждый из уровней нужно изменить имя каждого из них. Уровень 1-Выбор. проекта, Уровень 2-критерии, Уровень 3-названия проектов (рис. 1).

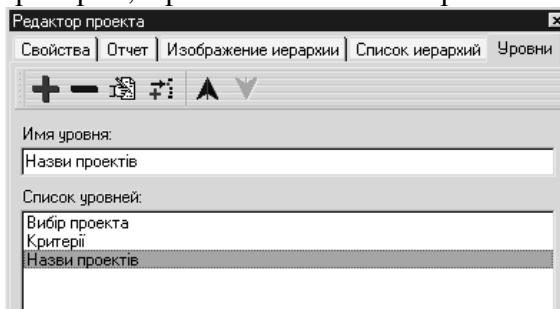


Рис. 1. Диалоговое окно "Редактор проекта".

7. Далее нужно нажать на "Выбор проекта", которое находится в списке дважды левой кнопкой мыши.

8. Появится окно "Редактор уровня".

9. Нужно щелкнуть на вкладке “Узлы”, и нажимая на красный знак “+” добавить узел, и изменить его имя на “Цель”.

10. Затем нужно нажать ОК и сделать так же с уровнями 2 и 3. Во второй уровень нужно добавить 3 узла и назвать их критериям условия.

11. Теперь нужно установить связь между каждым узлом. Для этого нужно вызвать окно “Редактор уровня” используя шаги 7 и 8.

12. Для того чтобы установить связи каждого из узлов нужно дважды щелкнуть на имени узла и в окне “Редактор узла” выбрать вкладку “Связи”.

13. Входных связей первого уровня не будет поэтому нужно выбрать вкладку “Исходящие” и выбрать нажав на стрелку возле выпадающего списка, выбрав в нем критерии.

14. В списке ниже появятся критерии, требуется выбрать все.

15. Так же нужно сделать с уровнем Критерии.

16. Установив необходимые связи мы выполнили первую часть по созданию проекта типа “Стоимость-эффективность” (рис. 2):

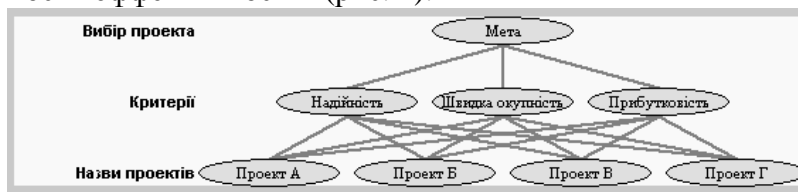


Рис. 2. Схема работы СППР (иерархия предпочтений).

17. Сохраняем проект выполняя команду “Файл” – “Сохранить”. Далее выполняем команду “Проект” – “Иерархия” – “Выбор иерархии” – “Иерархия издержек”.

18. Повторяем действия, описанные в пунктах 2-9.

19. Затем нужно нажать ОК и сделать так же с уровнями 2 и 3. Во второй уровень нужно добавить 3 узла и назвать их критериям из условия (сложность контроля, жаль н/с-цу, удаленность).

20. Повторяем действия, описанные в пунктах 11-15.

21. Сохраняем проект выполняя команду “Файл” – “Сохранить”. В результате получаем схему, изображенную на рис. 3:

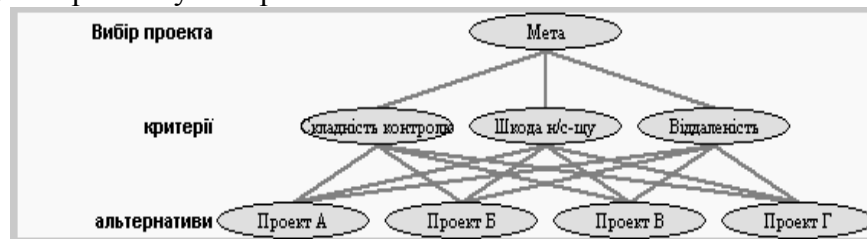


Рис. 3. Схема работы СППР (иерархия потерь)

Для проверки правильности проделанной работы нужно запустить процесс расчета командой “Проект” – “Расчет”, если все сделано верно, начнется процесс подсчета. В противном случае нужно исправить ошибки. Самая распространенная ошибка – недостаточность связей между узлами, что легко исправить добавив их или заменив другими.

Задания по вариантам:

1. Создать проект выбора мобильных телефонов.
2. Создать проект выбора телевизоров.
3. Создать проект выбора холодильников.
4. Создать проект выбора процессоров.
5. Создать проект выбора ОЗУ.
6. Создать проект выбора Винчестеров.
7. Создать проект выбора колонок.
8. Создать проект выбора магнитол.

9. Создать проект выбора приводов.
10. Создать проект выбора мониторов.
11. Создать проект выбора мышек.
12. Создать проект выбора клавиатур.
13. Создать проект выбора материнских плат.
14. Создать проект выбора принтеров.
15. Создать проект выбора модемов.
16. Создать проект выбора ксероксов.
17. Создать проект выбора стационарных телефонов.
18. Создать проект выбора охлаждающих систем (имеется в виду кулеры или водяное охлаждение).
19. Создать проект выбора тюнеров.
20. Создать проект выбора бесперебойников.

Лабораторная работа 5. Исследование с помощью системы "Выбор"

Тема: Работа с СППР "Выбор", средством создания проектов типа "Стоимость-эффективность". Расчеты, представление информации, выводы

Цель занятия:

1. Изучение возможностей проекта типа "Стоимость-эффективность" в программе "Выбор" как элемента моделирования СППР, завершение проекта предыдущего практического занятия

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения:

1. Отличие матрицы прямых сравнений в проектах "Проблема выбора" и "Стоимость-эффективность".

2. Дайте общую характеристику задачи "Стоимость-эффективность".

3. Перечислите классы задач, которые может решать проект "Стоимость-эффективность".

4. Преимущества и недостатки проекта "Стоимость-эффективность".

Ход работы

1. Изучить основные аспекты вычислений в СППР "Выбор".

2. Ознакомиться с интерфейсом и основными командами для проведения расчетов в СППР "Выбор".

3. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.

4. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

Практическое задание.

Произвести нужные вычисления и оформить выводы с предварительно сделанным проектом выбора проекта.

Ход выполнения работы:

1. Предварительный проект командой “Файл” – “Открыть”
2. Командой “Проект” – “Расчет”, вызываем окно, изображенное на рис. 4:

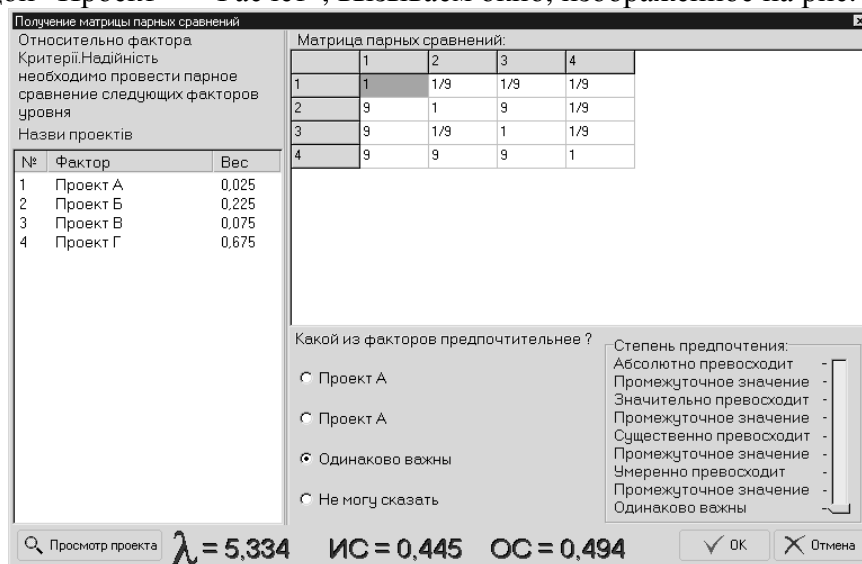


Рис. 4. Окно “Получение матрицы парных сравнений”

3. Данное окно представляет собой нечто вроде тестирующей программы оператором которого является матрица, которую нужно заполнить значениями. Для этого нужно нажимать на каждой ячейке под главной диагональю и отвечать на вопросы. Когда каждое поле будет заполнено нужно нажать ОК программа обработает информацию и выведет следующее окно (рис. 5):

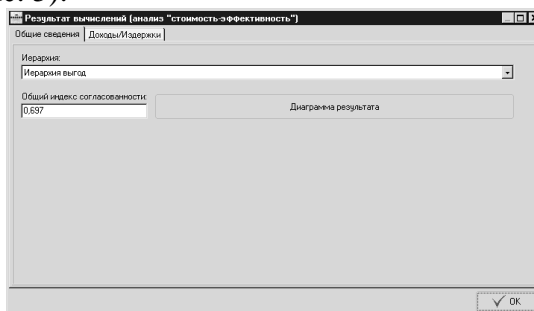


Рис. 5. Окно “Результат вычислений (анализ “стоимость-эффективность”)”.

4. Индекс согласованности показывает на то что данные которые Вы ввели не противоречивыми.

5. Далее нужно вывести диаграмму результата нажав на аналогичную клавишу в окне “Результат Вычислений”. Будет выведена диаграмма иерархии предпочтений (рис. 6). Для вывода иерархии потерь необходимо вернуться в окно “Результат вычислений (анализ “стоимость-эффективность”)” и в поле “Иерархия” выбрать параметр “Иерархия издержек”.

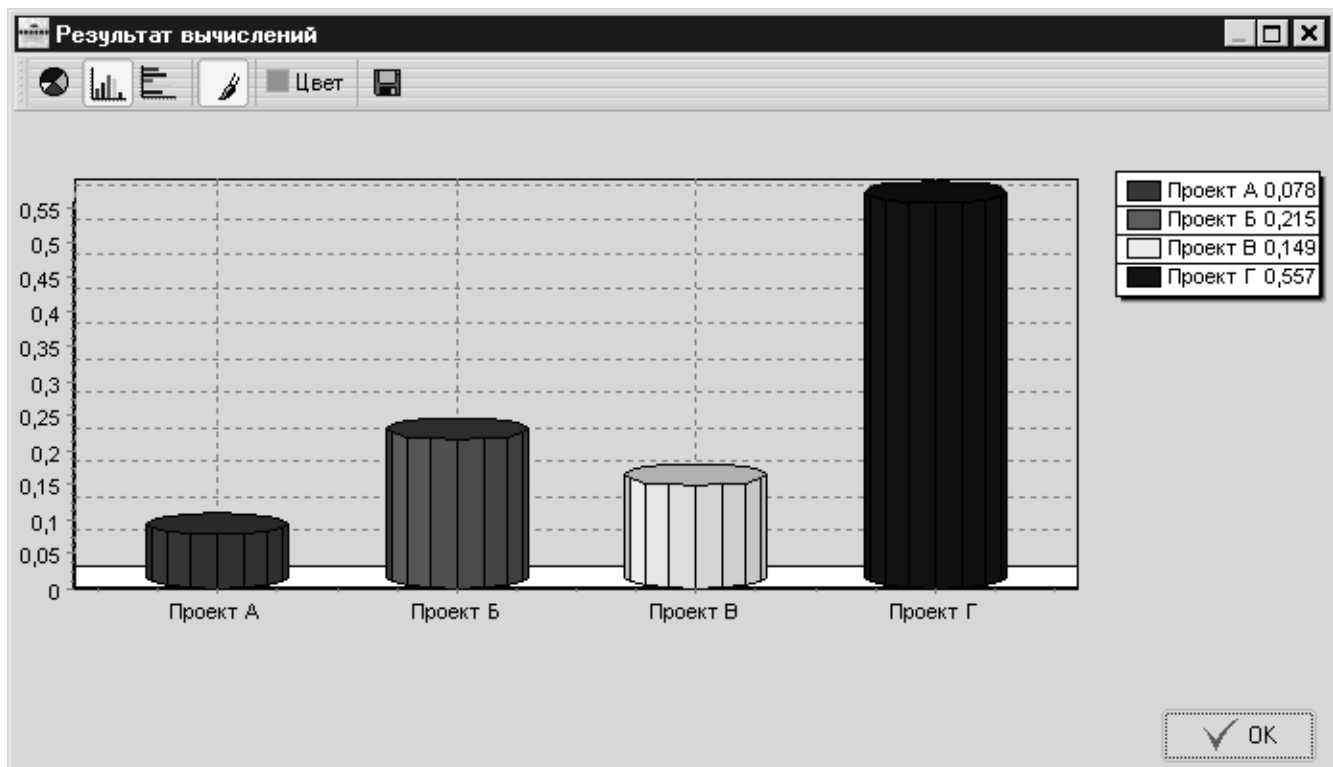


Рис. 6. Окно “Результат вычислений”

6. В данном окне можно выбрать вид диаграммы, цвет, можно также сохранить ее. Нужно сохранить проект. Диаграмма и расчеты сохраняются вместе с проектом. Задание к лабораторной работе брать с предыдущей работы.

Лабораторная работа 6. Исследование с помощью системы "Выбор"

Тема: Работа с СППР “Выбор”. Расчеты, представление информации, выводы

Цель занятия:

1. Изучение возможностей проекта типа “Поиск наилучшего решения” в программе “Выбор” как элемента моделирования СППР.

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения

1. Теория важности критериев.
2. Свёртка критериев. Однородность критериев.

3. Методы определения качественной важности критериев.
4. Определение количественной важности критериев.
5. Методы определения коэффициентов важности критериев.

Ход работы

1. Изучить основные аспекты вычислений в СППР «Выбор».
2. Ознакомиться с интерфейсом и основными командами для проведения расчетов в СППР «Выбор».
3. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
4. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР «Выбор» в СППР, описание примеров, иллюстративный материал.

С помощью программы «Выбор» попытаемся решить следующую задачу. Нам необходимо произвести отбор кандидатов на освободившуюся должность заместителя начальника отдела информатизации из числа сотрудников отдела. Кандидатов будем оценивать по нескольким критериям:

- стаж работы в организации,
- ответственность,
- образование,
- коммуникабельность.

Мы имеем 4-х претендентов на эту должность.

Таблица 1 – Критерий кандидатов

Ф.И.О.	Критерий			
	Стаж работы	Ответственность	Коммуникабельность	Образование
Иванов	5	Очень ответственный	Коммуникабельный	Высшее техническое
Петров	2	Достаточно ответственный	Замкнут (не коммуникабельный)	Высшее гуманитарное
Сидоров	1	Не ответственный	Очень коммуникабельный	Средне специальное
Потапов	3	ответственный	Достаточно коммуникабельный	Незаконченное высшее

С помощью программы попытаемся проанализировать, кто из претендентов наиболее подходит.

Сначала нам необходимо ввести в программу данные по критериям и фамилии претендентов. (Рис. 3)

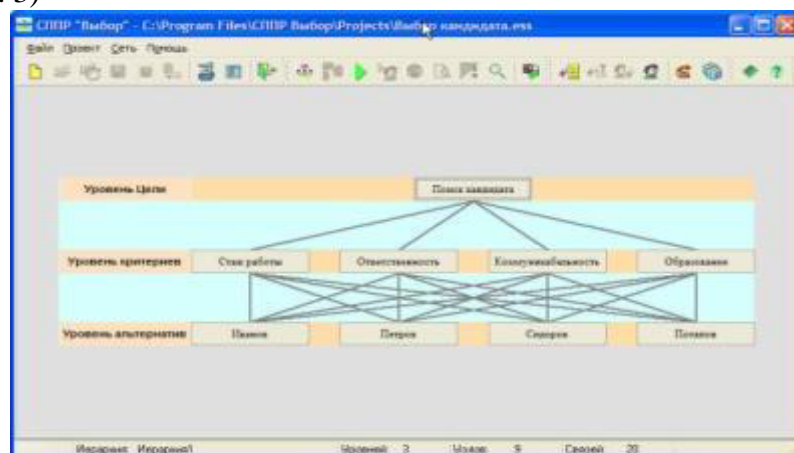


Рисунок 3 – Данные по критериям

Затем мы запускаем выполнение вычислений, где нам необходимо относительно каждого уровня произвести оценку нескольких факторов, тем самым расставив предпочтения (рис. 4)

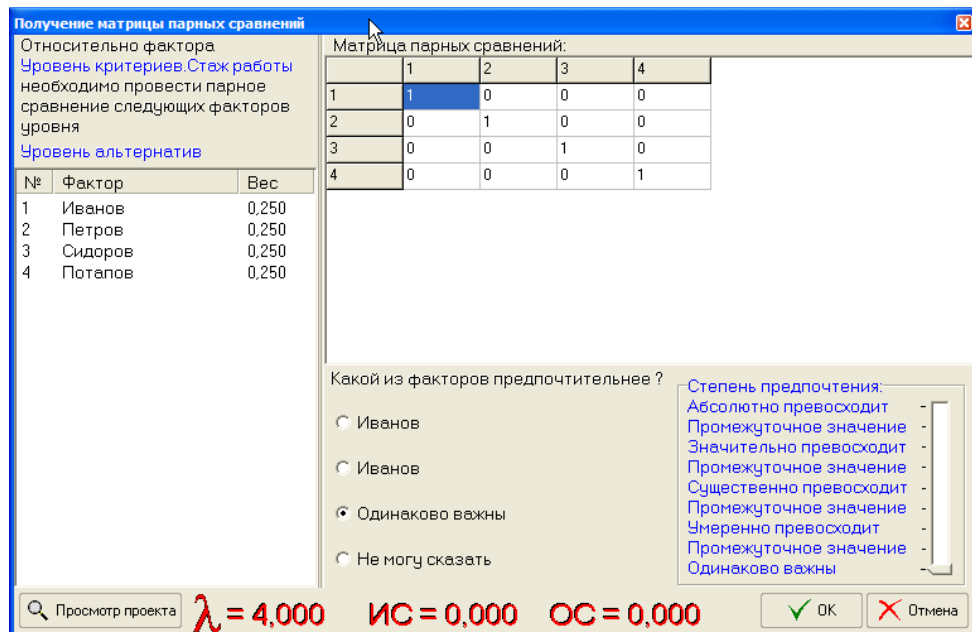


Рисунок 4 – Расчет предпочтений

Для каждого критерия производим оценку, допустим по стажу работы Иванов имеет большее предпочтение так как он проработал на нашем предприятии дольше Петрова. (рис. 5)

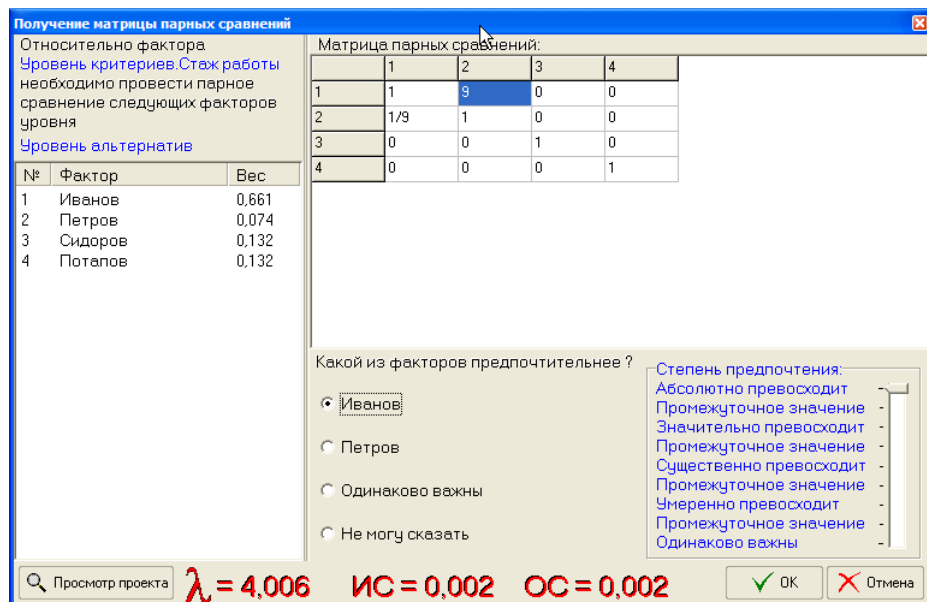


Рисунок 5 – Оценка критериев

Полученные матрицы парных сравнений по критерию стаж работы (рис. 6):

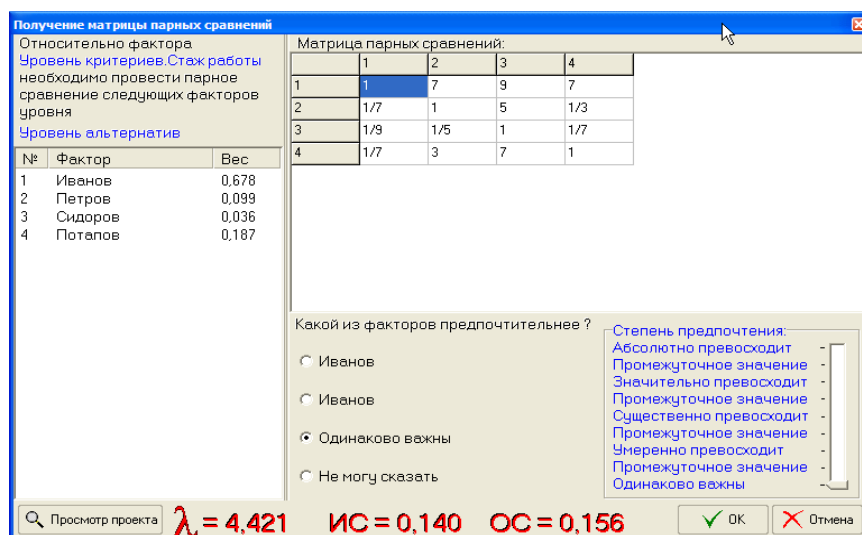


Рисунок 6 – Сравнение по критерию стажа

Иванов проработал дольше всех на предприятии, поэтому ему достается самая высокая оценка, а Сидоров проработал мало, у него самая маленькая.

Полученные матрицы парных сравнений по критерию ответственность (рис. 7):

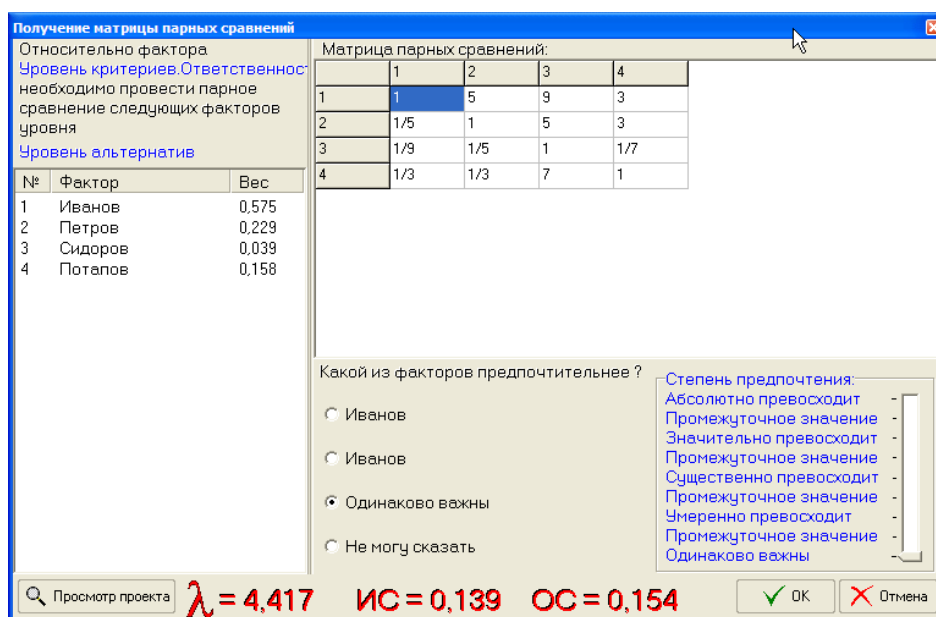


Рисунок 7 – Сравнение по критерию ответственности

Из этой матрицы мы видим, что наибольшая оценка опять достается Иванову, он признан самым ответственным всегда выполняя поручения, Сидоров же наоборот очень часто срывает сроки и относился халатно, поэтому у него самая маленькая оценка.

Полученные матрицы парных сравнений по критерию коммуникабельность (рис. 8):

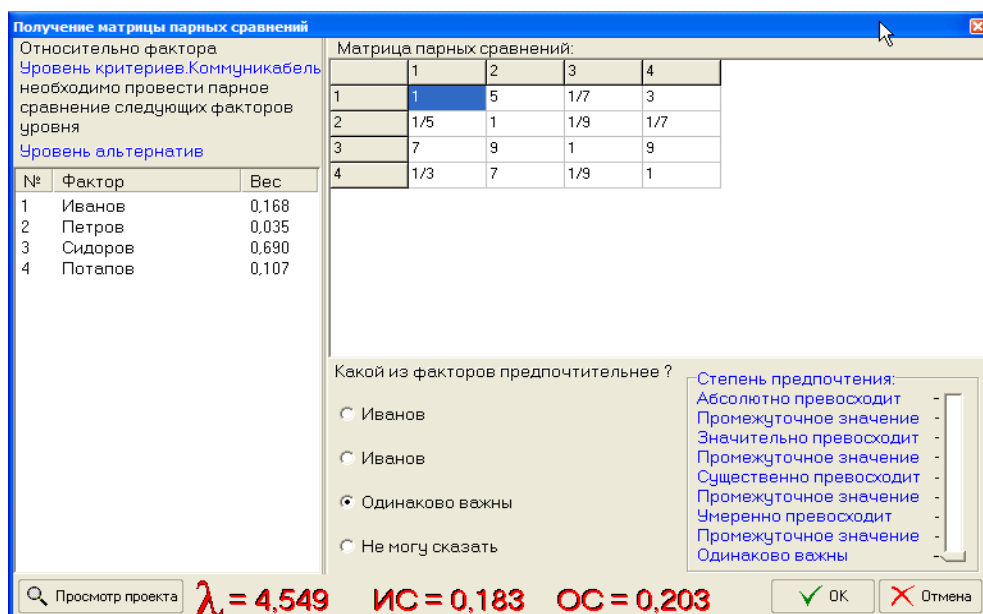


Рисунок 8 – сравнение по критерию коммуникабельности

Здесь мы видим очень интересную картину, по нашим наблюдениям наиболее коммуникабельным признан Сидоров, который по предыдущим критериям был аутсайдером, а вот Петров оказался позади всех, и признан замкнутым человеком.

Комментирую эту матрицу мы должны вспомнить постановку задачи, нам нужен заместитель начальника отдела информатизации, т.е. это должен быть человек хорошо разбирающийся в работе отдела, а значит техник по образованию, но с другой стороны это руководящая должность и возможно человек долго проработавший в отделе хоть и с гуманитарным образованием должен иметь равные шансы. Поэтому как мы видим на рис. 9, Иванову и Петрову проставлены одинаковые оценки, меньше всего у Сидорова т.к. руководитель должен иметь высшее образование.

Полученные матрицы парных сравнений по критерию образование (рис. 9):

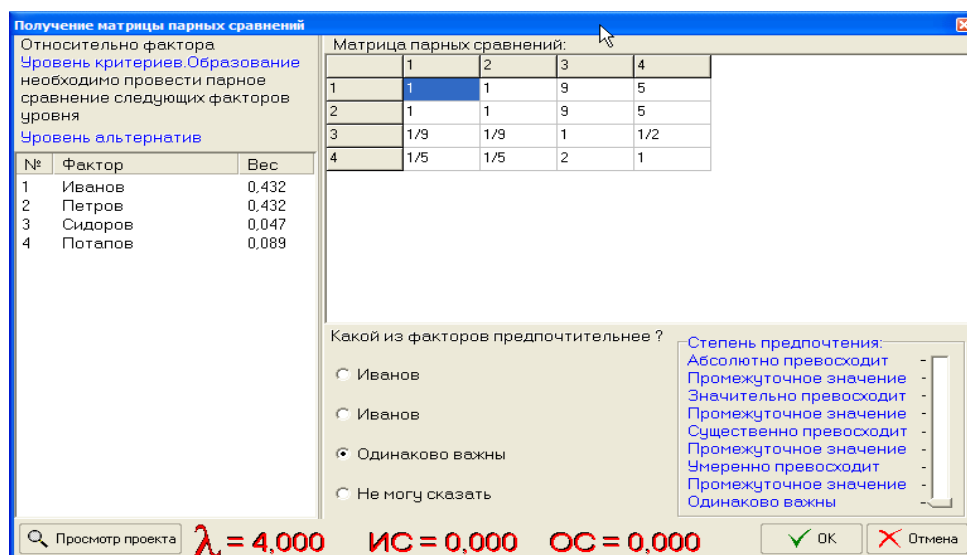


Рисунок 9 – Сравнение по критерию образования

Что для нас является важнее? Стаж, образование, коммуникабельность или ответственность? Думаю, уровнем шансы, нам нужен кандидат в равной степени удовлетворяющий всем параметрам. Расставляем приоритеты поровну. (рис. 10)

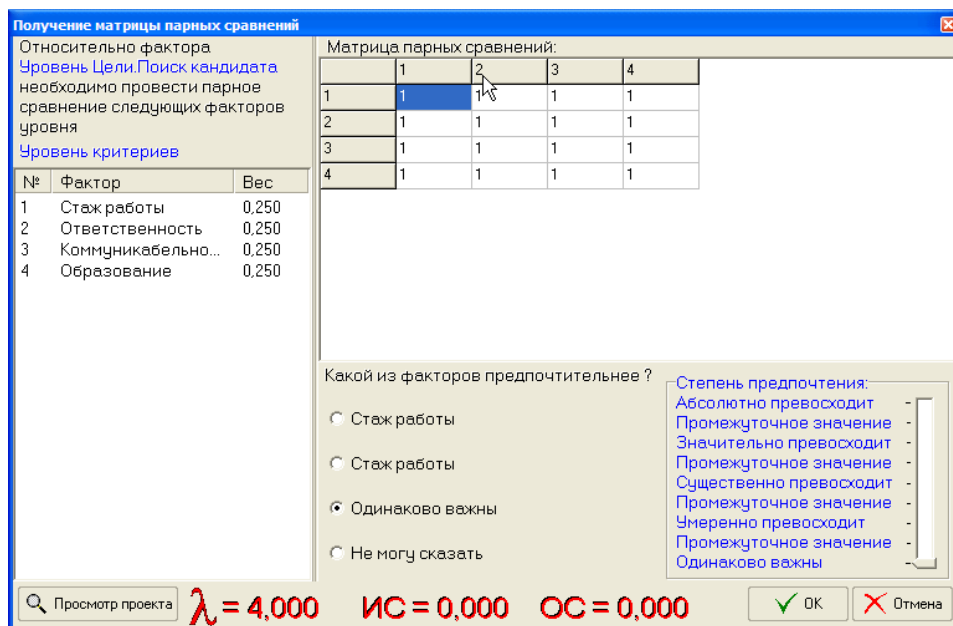


Рисунок 10 – Общие критерий кандидатов

Вычисления закончены, получаем результат (рис. 11):

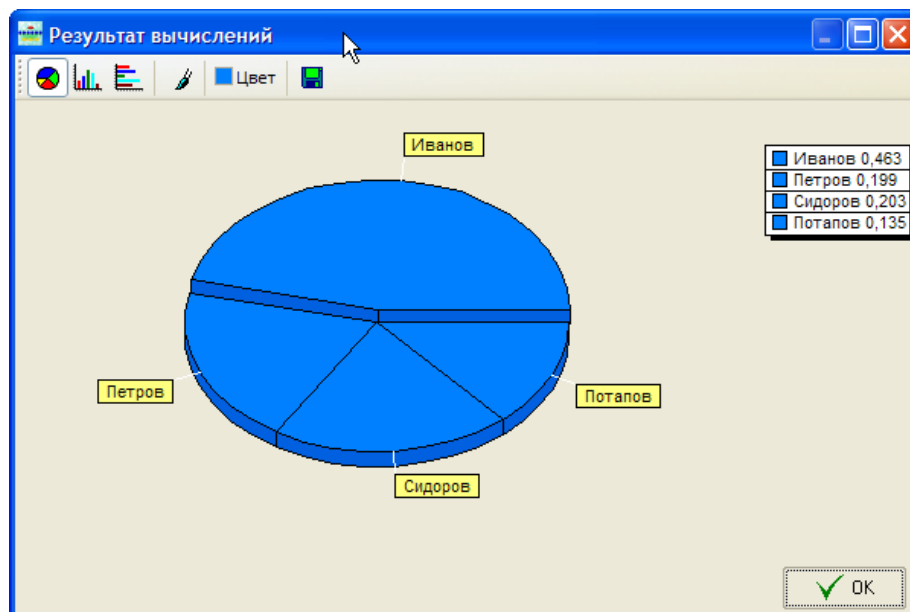


Рисунок 11-Получение результатов

Очевидно, что наиболее подходящим кандидатом на должность заместителя начальника отдела является Иванов.

Задание 1

С помощью программы «Выбор» решить следующую задачу. Нам необходимо произвести отбор кандидатов на освободившуюся должность старосты из числа студентов вашей группы. Кандидатов оценивать по нескольким критериям:

- успеваемость,
- ответственность,
- образование,
- коммуникабельность.

Произвести нужные вычисления и оформить выводы с предварительно сделанным проектом.

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

6.1.1. Основная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л1.1	Петров, А. Е.	Математические модели принятия решений: учебно-методическое пособие	Москва: Издательский Дом МИСиС, 2018	http://www.iprbookshop.ru/78572.html
Л1.2	Муромцев, Д. Ю., Шамкин, В. Н.	Методы оптимизации и принятие проектных решений: учебное пособие для магистрантов по направлению 11.04.03	Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2015	http://www.iprbookshop.ru/63866.html
Л1.3	Горелик, В. А.	Теория принятия решений: учебное пособие для магистрантов	Москва: Московский педагогический государственный университет, 2016	http://www.iprbookshop.ru/72518.html
6.1.2. Дополнительная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л2.1	Бережная, О. В., Бережная, Е. В.	Методы принятия управленческих решений: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2015	http://www.iprbookshop.ru/62960.html
	Авторы,	Заглавие	Издательство, год	Адрес
Л2.2	Казанская, О. В., Юн, С. Г., Альсова, О. К.	Модели и методы оптимизации. Практикум: учебное пособие	Новосибирск: Новосибирский государственный технический университет, 2012	http://www.iprbookshop.ru/45397.html
6.1.3. Методические разработки				
	Авторы,	Заглавие	Издательство, год	Адрес
Л3.1	Палинчак, Н. Ф., Ярославцева, В. Я.	Системный анализ, оптимизация и принятие решений: методические указания и задания для самостоятельной работы	Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2014	http://www.iprbookshop.ru/55156.html
Л3.2	Артюхин Г. А.	Теория систем и системный анализ. Практикум принятия решений: Учебное пособие	Казань: Казанский государственный архитектурно-строительный университет, ЭБС АСВ, 2016	http://www.iprbookshop.ru/73321.html



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Экономико-математические модели управления»
для студентов направления подготовки
09.04.02 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Экономико-математические модели управления» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1. Исследование с помощью системы "Выбор"

Лабораторная работа 2. Исследование с помощью системы "Выбор"

Лабораторная работа 3. Исследование с помощью системы "Выбор"

Лабораторная работа 4. Исследование с помощью системы "Выбор"

Лабораторная работа 5. Исследование с помощью системы "Выбор"

Лабораторная работа 6. Исследование с помощью системы "Выбор"

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины - развить системное мышление у обучающихся путем детального анализа подходов к математическому моделированию и сравнительного анализа разных типов моделей. Ознакомить обучающихся с математическими свойствами методов и моделей оптимизации, которые могут использоваться при анализе и решении широкого спектра задач. Выработать у обучающихся навыки проведения численных исследований математических моделей и анализа результатов вычислений. Научить выбирать наиболее перспективное управляющее решение.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Изучив данный курс, студент должен:

Знать:

основы моделирования управленческих решений;

математические модели и информационные технологии процесса поддержки принятия решений;

многокритериальные методы поддержки принятия решений;

основные технологии информационной поддержки процесса поддержки принятия решений;

классификацию систем поддержки принятия решений и особенности используемых инструментальных средств;

современные методы и средства поддержки принятия решений в различных интеллектуальных системах,

принципы их рационального выбора в зависимости от особенностей процесса поддержки принятия решений.

Уметь:

осуществлять постановку конкретных задач поддержки принятия решений, выбирать адекватные математические и инструментальные средства их решения;

решать задачи, связанные с различными этапами подготовки и принятия решений в инструментальных системах

Владеть:

навыками формулирования требований к методам и моделям поддержки принятия решений;

навыками разработки отдельных их элементов;

навыками практического использования моделей и методов поддержки принятия решений;

навыками аналитического обоснования вариантов решений с использованием систем поддержки принятия решений.

Реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа 1. Исследование с помощью системы "Выбор"

Цель занятия:

1. Понятие СППР. Эволюция информационных технологий и информационных систем. Усвоить основные теоретико-информационные понятия учебной дисциплины изучить этапы развития информационной технологии и информационных систем.

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения

1. Дайте определение понятию конкурентные гонки?
2. Что такое информация?
3. Определите содержание СППР?
4. Чем отличаются данные от информации?
5. Основные характеристики данных?

Ход работы

1. Информация и данные.
2. Развитие информационных технологий.
3. Перспективные средства и направления развития информационных систем.
4. Основные понятия систем поддержки принятия решений.

Задача 1. Привести по три определения каждого понятия: “информация”, “данные”, “система поддержки принятия решений” (СППР).

Задача 2. Выделить критерии отбора альтернативных вариантов, которые, по вашему мнению, должны входить в состав СПР выбранной тематики. (например инвестиционные проекты: прибыль, срок окупаемости, и прочее). Также нужно выделить главные и второстепенные критерии. обосновать свой выбор. Тема определяется в соответствии с номером студента в академическом журнале (см темы разработки сппр).

Задача 3. Привести несколько существенных преимуществ применения СППР в выбранной области. Ответ обоснуйте.

Темы разработки сппр:

1. Создание проекта выбора ПК
2. Создание проекта выбора ТВ
3. Создание проекта выбора монитора
4. Создание проекта выбора микроволновой печи
5. Создание проекта выбора автомобиля
6. Создание проекта выбора магнитолы
7. Создание проекта выбора принтера
8. Создание проекта выбора сканера
9. Создание проекта выбора плоттера
10. Создание проекта выбора материнской платы
11. Создание проекта выбора процессора
12. Создание проекта выбора модема
13. Создание проекта выбора мобильного телефона
14. Создание проекта выбора программного обеспечения (по)
15. Создание проекта выбора винчестера (НЖМД)
16. Создание проекта выбора стационарного телефона
17. Создание проекта выбора DVD-проигрывателя
18. Создание проекта выбора интернет-провайдера
19. Создание проекта выбора видеокарты
20. Создание проекта выбора факсимильного аппарата

Лабораторная работа 2. Исследование с помощью системы "Выбор"

Тема: работа с СППР "Выбор". Основные функции, приемы и возможности

Цель занятия:

1. Ознакомиться с основными командами и получить базовые навыки при работе с СППР "Выбор"

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения

1. Назовите основные возможности СППР "Выбор";

2. Перечислите основные команды, которые потребуются для создания нового работоспособного проекта.
3. Перечислите основные возможности окна разработки проектов.
4. Перечислите основные преимущества и недостатки СППР “Выбор”.

Ход работы:

1. Ознакомиться с основными принципами метода анализа иерархий.
2. Изучить основные аспекты интерфейса СППР “Выбор”.
3. Ознакомиться с интерфейсом и основными командами для пользования инструментами доступными в СППР “Выбор”.
4. Познакомиться с основными командами СППР “Выбор”.
5. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
6. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

Основные аспекты интерфейса СППР "Выбор".

Система поддержки принятия решений (СППР) “Выбор”- аналитическая система, основанная на методе анализа иерархий (МАИ), является простым и удобным средством, которое поможет структурировать проблему, построить набор альтернатив, выделить факторы, характеризующие их, задать значимость этих факторов, оценить альтернативы по каждому из факторов, найти неточности и противоречия в суждениях лица принимающего решение (ЛПР эксперта), проранжировать альтернативы, провести анализ решения и обосновать полученные результаты. Система опирается на математически обоснованный метод анализа иерархий Томаса Саати.

Клиентское применение СППР “Выбор” обладает интуитивным пользовательским интерфейсом. Главное окно-это инструмент работы над проектом, позволяющий просматривать и редактировать выбранную иерархию проекта.

Основные компоненты интерфейса СППР " Выбор”

Панели инструментов.

С помощью панелей инструментов осуществляется быстрый доступ к основным инструментам приложения сосредоточенным в главном меню. Каждому пункту меню соответствует кнопка быстрого запуска на панели инструментов.

Подсказки.

Практически все элементы окон снабжены подсказками, которые появляются при небольшой задержке курсора мыши на необходимом элементе. Более подробная информация обо всех инструментах приложения содержится в помощи, сосредоточенной в главном меню Помощь.

Контекстно-зависимая помощь

СППР “Выбор” обладает мощной справочной системой по каждому инструменту приложения. Более 300 пунктов помощи помогут вам быстрее разобраться как использовать тот или иной элемент приложения. Помощь также предоставляет поиск информации по ключевым словам и фразам.

Помощь по диалоговым окнам

Все диалоговые окна снабжены справочной информацией. Нажмите клавишу F1 для вызова справки по текущему окну. Панели инструментов полностью копируют элементы главного меню. Каждому элементу главного меню соответствует панель инструментов:

* Файл-открытие, сохранение, закрытие проектов, создание новых, настройка приложения, принтера для печати, а также закрытие приложения.

• Проект - работа с иерархиями, произведение расчетов, получение отчетов, редактирование свойств проекта.

* Сеть-отсылка выбранным экспертам текстовых сообщений, проектов, открытие пришедших сообщений, просмотр списка сетевых событий.

* Помощь-вызов справки и окна информации о приложении.

В главном окне приложения для быстрого вызова некоторых инструментов можно использовать следующие горячие клавиши:

Файл

Ctrl + n-новый
F3-Открыть
F2-Сохранить
Ctrl + f2-Сохранить как
F10-Закрыть
F4-настройка принтера
Ctrl+o - Опции

Проект

Ctrl + i - информация о готовности
Ctrl + c-расчет
Ctrl + alt + c - сетевой расчет
Ctrl + r-результаты расчетов
Ctrl + p-просмотр отчета
Ctrl + t-изменить типа
Ctrl+v - Режим просмотра
Shift+Ctrl + P - Свойства

Сеть

F5-текстовое сообщение
F6-послать проект
F10-выбрать экспертов
F9-Эксперты
F11 - Окно сетевых сообщений
F12-окно с пакетами

Помощь

F1-Информация
Ctrl + f1-о программе

В окнах со списками, если фокус ввода находится на списке, то становятся доступными следующие горячие клавиши:

- Insert или "+" - добавление записи.
- * Delete или "-" - удаление текущей записи.

Практическое задание.

Создать проект выбора автомобиля имея такую информацию:

Марки автомобилей 5-10 шт.

Критерии оценки:

1. Мощность двигателя (мощный)
2. Цена (Высокая)
3. Качество (Высокое)
4. Ведомость марки (Известная)
5. Комфортность (Комфортная)
6. Стильность (Стильная)

Создание проекта

1. Командой “Файл” – “Новый...” – “Простой проект” создать новый проект.
2. Щелкнуть на прямоугольнике по центру окна правой кнопкой мыши, вызвав при этом контекстное меню и в нем выбрать команду “Свойства проекта”.
3. Выбрать вкладку "Уровни".

4. Щелкнуть на знаке “+” 3 раза.
5. В списке появится 3 уровне (рис. 1).

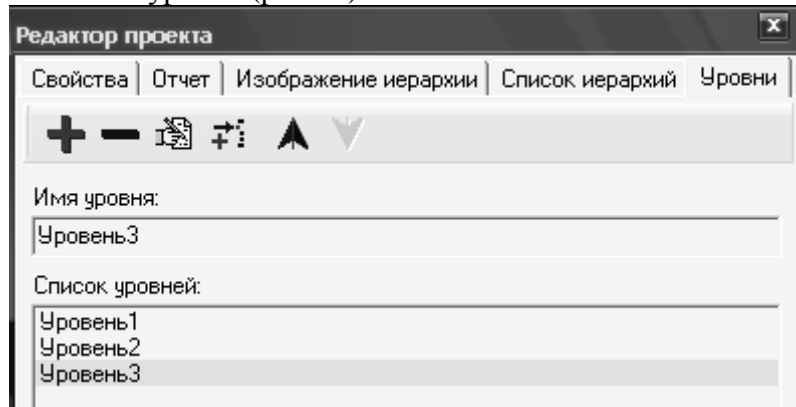


Рис. 1. Окно “Редактор проекта”.

6. Нажимая на каждый из уровней нужно изменить имя каждого из них. Уровень1 – Покупка автомобиля, Уровень2 – Критерии, Уровень3 – Марки авто.

7. Далее нужно нажать на “Покупка автомобиля”, которое находится в списке дважды левой кнопкой мыши.

8. Появится окно “Редактор уровня3”.

9. Нужно щелкнуть на вкладке “Узлы3, и нажимая на красный знак “+” добавить узел, и изменить его имя на “Цель”.

10. Затем нужно нажать ОК и сделать так же с уровнями 2 и 3. Во второй уровень нужно добавить 6 узлов и назвать их критериям условия, в третий уровень нужно добавить минимум 5 узлов и вписать любые марки машин так, чтобы каждой машине соответствовали по крайней мере 2 критерии.

Замечание !!! Для облегчения понимания программы рекомендуем писать только положительные качества предметов цели, то есть если идет речь о выборе машин и о критериях “цена” нужно в критерии добавить узел “высокая цена”. Если авто с низкой ценой между узлом марки авто и узлом “высокая цена” просто не делать связь.

11. Теперь нужно установить связь между каждым узлом. Для этого нужно вызвать окно “Редактор уровня” используя шаги 7 и 8. Окно СППР “Выбор” к установлению связей показаны на рис. 2:

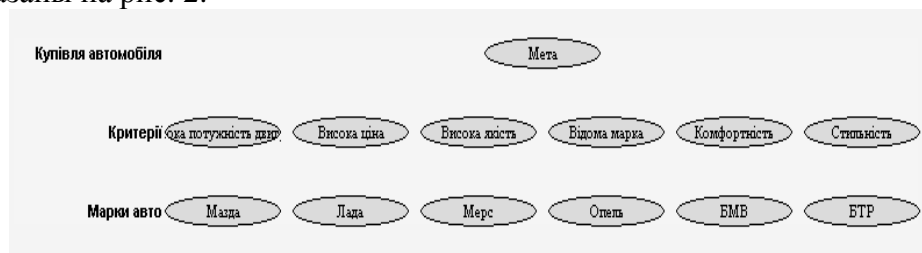


Рис. 2. Окно СППР “Выбор” к установлению связей.

12. Для того чтобы установить связи каждого из узлов нужно дважды щелкнуть на имени узла и в окне “Редактор узла” выбрать вкладку “Связи”.

13. Входных связей первого уровня не будет поэтому нужно выбрать вкладку “Исходящие” и выбрать нажав на стрелку возле выпадающего списка, выбрав в нем критерии.

14. В списке ниже появятся критерии, требуется выбрать все.

15. Так же нужно сделать с уровнем Критерии.

16. Установив необходимые связи получим готовый для расчетов проект (рис. 3.).

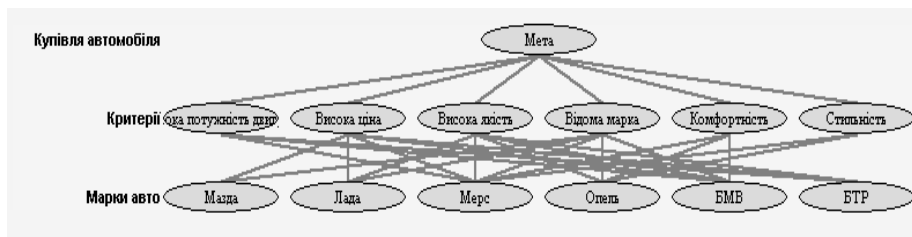


Рис. 3. Окно СППР “Выбор” после установления связей.

Для проверки правильности проделанной работы нужно запустить процесс расчета командой “Проект” – “Расчет”, если все сделано верно, начнется процесс подсчета. В противном случае нужно исправить ошибки. Самая распространенная ошибка-недостаточность связей между узлами, что легко исправить добавив их или заменив другими.

Далее нужно сохранить проект командой “Файл” – “Сохранить”.

Варианты заданий:

1. Создать проект выбора мобильных телефонов
2. Создать проект выбора телевизоров.
3. Создать проект выбора холодильников.
4. Создать проект выбора процессоров.
5. Создать проект выбора ОЗУ.
6. Создать проект выбора Винчестеров.
7. Создать проект выбора колонок.
8. Создать проект выбора магнитол.
9. Создать проект выбора проводов.
10. Создать проект выбора мониторов.
11. Создать проект выбора мешков.
12. Создать проект выбора клавиатур.
13. Создать проект выбора материнских плат.
14. Создать проект выбора принтеров.
15. Создать проект выбора модемов.
16. Создать проект выбора ксероксов.
17. Создать проект выбора стационарных телефонов.
18. Создать проект выбора охлаждающих систем (имеется в виду кулеры или водяное охлаждение).
19. Создать проект выбора тюнеров.
20. Создать проект выбора бесперебойников.

Лабораторная работа 3. Исследование с помощью системы "Выбор"

Тема: работа с СППР "Выбор". Расчеты, представление информации, выводы.

Цель занятия:

1. Ознакомиться с основными командами и получить базовые навыки при работе с СППР “Выбор”.

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения:

1. Охарактеризовать принципы работы матрицы парных сравнений.
2. Охарактеризовать окно “Результат вычислений”.
3. Перечислите команды, которые нужно выполнить для изменения вида диаграммы.
4. Для чего нужен индекс согласованности.
5. Для чего нужны веса у исследуемых альтернатив?

Ход работы:

1. Изучить основные аспекты вычислений в СППР “Выбор”.
2. Ознакомиться с интерфейсом и основными командами для проведения вычислений в СППР “Выбор”.
3. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
4. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

задание

Провести нужные расчеты и оформить выводы с предварительно сделанным проектом выбора машины.

Ход выполнения работы

1. Открыть предыдущий проект командой "Файл" - "Открыть".
2. Командой “Проект” – “Расчет”, вызвать окно, изображенное на рис. 1:

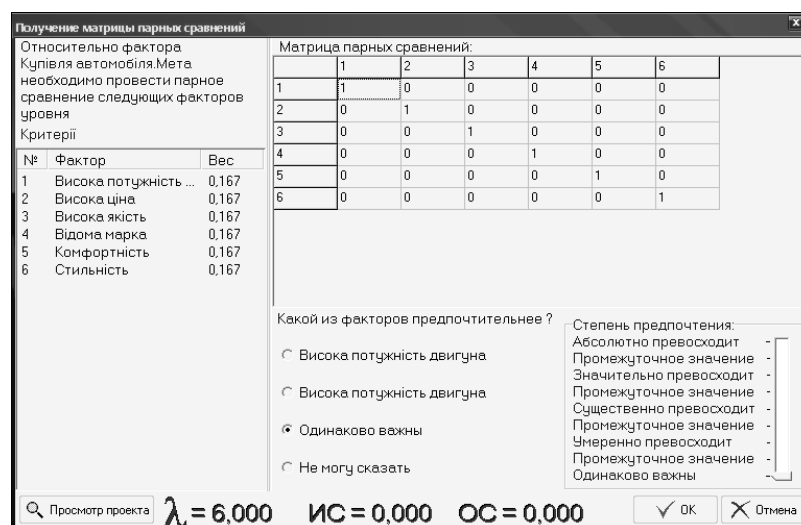


Рис. 1. Окно «Получение матрицы парных сравнений»

3. Данное окно представляет собой нечто вроде тестирующей программы оператором которого является матрица, которую нужно заполнить значениями. Для этого нужно нажимать на каждой ячейке под главной диагональю и отвечать на вопросы. когда

каждое поле будет заполнено нужно нажать ОК программа обработает информацию и выведет следующее окно (рис. 2).

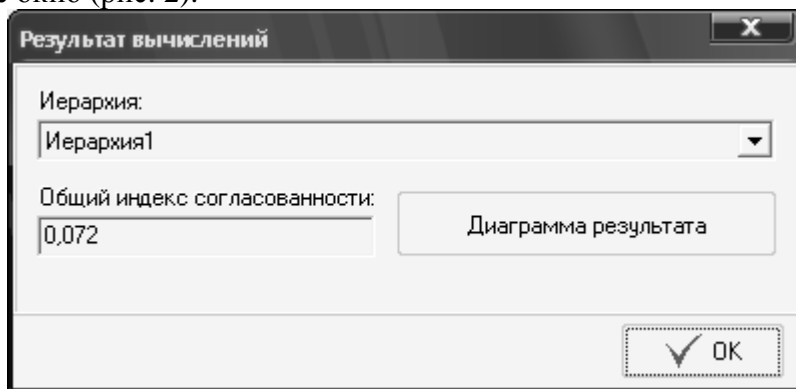


Рис. 2. Окно “Результат вычислений (анализ “Проблема выбора ”)”.

4. Индекс согласованности показывает на то что данные которые Вы ввели не противоречивыми.

5. Далее нужно вывести диаграмму результата нажав на аналогичную клавишу в окне “Результат Вычислений (рис. 3):

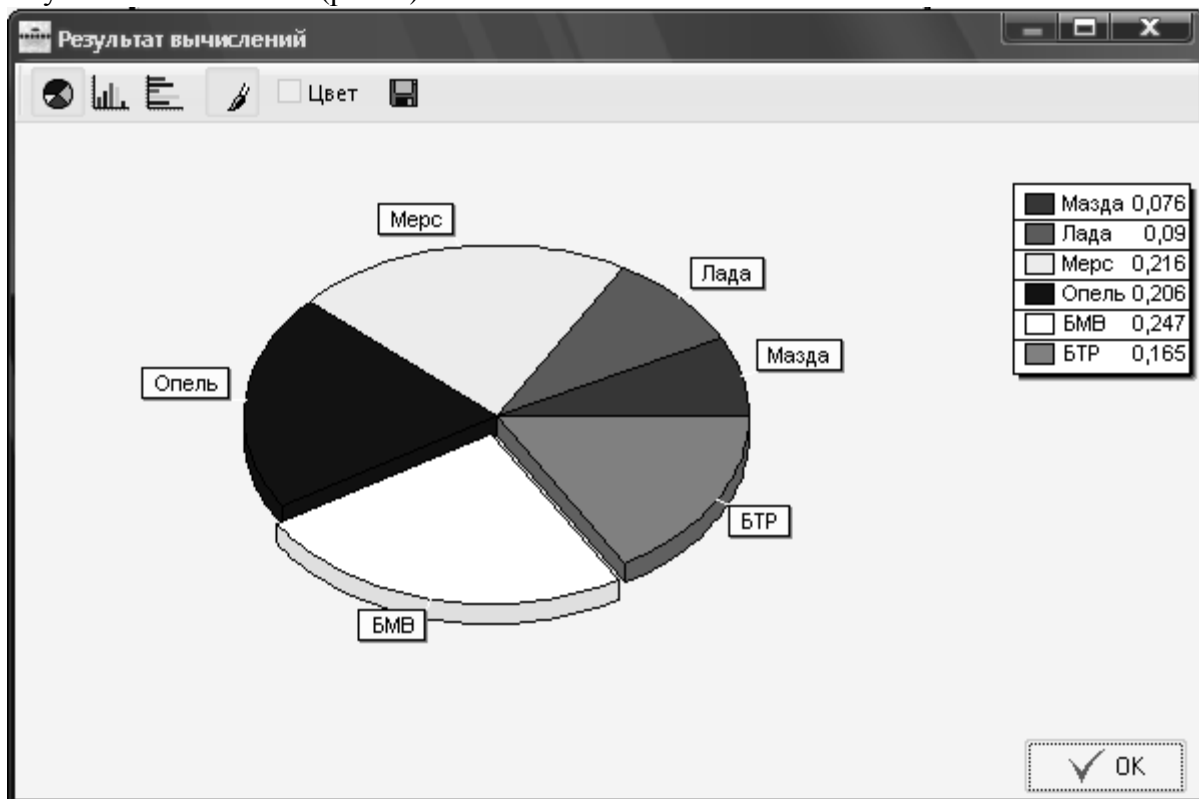


Рис. 3. Окно “Результат Вычислений”

6. В данном окне можно выбрать вид диаграммы, цвет, можно также сохранить ее. Нужно сохранить проект. Диаграмма и расчеты сохраняются вместе с проектом.

Лабораторная работа 4. Исследование с помощью системы "Выбор"

Тема: работа с СППР "Выбор". Создание проекта типа “Стоимость-эффективность”

Цель занятия:

- 1.Изучение возможностей проекта типа “Стоимость-эффективность” в программе «Выбор» как элемента моделирования СППР
2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;
УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения:

1. Чем отличаются проекты “Проблема выбора” и “Стоимость – эффективность”?
2. Для чего нужны уровни критериев и альтернатив?
3. Охарактеризуйте структуру иерархии в СППР “Выбор”.
4. Преимущества и недостатки СППР “Выбор”.

Ход работы

1. Проработать теоретический материал, рекомендации по работе с программой.
2. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
3. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

Постановка задачи и суть проекта типа “Стоимость-эффективность”.

Проекта типа “Стоимость-эффективность”.- это проект, состоящий из двух иерархий, иерархии выгод и иерархии затрат, которые впоследствии необходимо будет между собой ранжировать (сравнивать).

Например, рассматривается задача принятия решения по множеству планов или каких-либо проектов. Классический подход основан на оценке каждого проекта с точки зрения затрат (то есть, сколько необходимо сделать инвестиций для реализации данного проекта) и с точки зрения доходов, которые можно получить при их реализации. Сравнение альтернативных проектов сводится к сравнению объемов доходов из расчета на единицу ресурса (т. е. расходов). Этот метод известен как анализ “Стоимость-эффективность”.

Для примера можно предложить задачу по решению целесообразности выбора определенного проекта. Решение задачи ранжирования проектов а, в, С при традиционном подходе может быть сведено в табл. 1.

Таблица 1. Классический подход к задаче “Стоимость-эффективность”

Проект	Расходы	Доходы	Доходы/ расходы	Ранжирования
А	500	1000	2	3
В	250	750	3	1
С	600	1300	2,1	2

При решении этой задачи возникают следующие особенности:

□ отношение доходов к затратам, оценивается в стоимостном выражении, по сути не является объективной мерой качества проекта: неясно, как, например, оценивать в

деньгах выгоды и затраты неосязаемых количественно показателей (то есть проблема измерения качественных факторов);

□ известно также, что доходы и расходы распределяются по многим сферам - социальным, экономическим, политическим, управленческим и их взаимосвязь влияет на оценку альтернатив.

Применение СППР позволяет снять эти проблемы. В этом случае нужно построить две иерархии: одну для затрат, другую для выгод с одними и теми же альтернативами на нижнем уровне. Таким образом, получают два вектора приоритетов - доходов и расходов. Затем вычисляют отношения доходов к расходам для каждой альтернативы. Наибольшее значение из этих отношений и определяет лучший проект.

В целом решение задач типа "Стоимость-эффективность" происходит в следующем порядке:

- а) правильная формулировка цели;
- б) построение иерархии выгод (все те же действия, что и в п. 1, но отталкиваться лишь от выгод, не обращает внимания на затраты);
- в) построение иерархии издержек (все те же действия, что и в п. 1, но отталкиваться лишь от издержек, не обращающих внимания на выгоды).

Пример решения.

Выбор темы разработки СППР происходит согласно номера студенте в академическом журнале.

Создать СППР для выбора инвестиционного проекта имея такую информацию:

Количество проектов-4.

Критерии оценки (позитивные или выгоды):

7. Надежность;
8. Быстрая окупаемость;
9. Доходность.

Критерии оценки (отрицательные, или расходы):

1. Сложность контроля;
2. Жаль н/с-щу;
3. Удаленность.

Создание проекта

1. Командой "Файл" – "Новый..." – "Простой проект" создать новый проект.
2. Щелкнуть на прямоугольнике по центру окна правой кнопкой мыши, вызвав при этом контекстное меню и в нем выбрать команду "Свойства проекта".
3. Выбрать вкладку "Уровни".
4. Щелкнуть на знаке "+" 3 раза.
5. В списке появится 3 уровне.
6. Нажимая на каждый из уровней нужно изменить имя каждого из них. Уровень 1-Выбор. проекта, Уровень 2-критерии, Уровень 3-названия проектов (рис. 1).

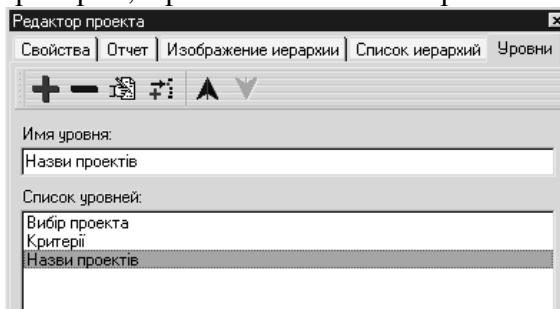


Рис. 1. Диалоговое окно "Редактор проекта".

7. Далее нужно нажать на "Выбор проекта", которое находится в списке дважды левой кнопкой мыши.

8. Появится окно "Редактор уровня".

9. Нужно щелкнуть на вкладке “Узлы”, и нажимая на красный знак “+” добавить узел, и изменить его имя на “Цель”.

10. Затем нужно нажать ОК и сделать так же с уровнями 2 и 3. Во второй уровень нужно добавить 3 узла и назвать их критериям условия.

11. Теперь нужно установить связь между каждым узлом. Для этого нужно вызвать окно “Редактор уровня” используя шаги 7 и 8.

12. Для того чтобы установить связи каждого из узлов нужно дважды щелкнуть на имени узла и в окне “Редактор узла” выбрать вкладку “Связи”.

13. Входных связей первого уровня не будет поэтому нужно выбрать вкладку “Исходящие” и выбрать нажав на стрелку возле выпадающего списка, выбрав в нем критерии.

14. В списке ниже появятся критерии, требуется выбрать все.

15. Так же нужно сделать с уровнем Критерии.

16. Установив необходимые связи мы выполнили первую часть по созданию проекта типа “Стоимость-эффективность” (рис. 2):

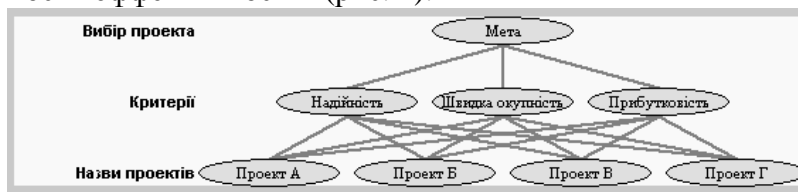


Рис. 2. Схема работы СППР (иерархия предпочтений).

17. Сохраняем проект выполняя команду “Файл” – “Сохранить”. Далее выполняем команду “Проект” – “Иерархия” – “Выбрать иерархию” – “Иерархия издержек”.

18. Повторяем действия, описанные в пунктах 2-9.

19. Затем нужно нажать ОК и сделать так же с уровнями 2 и 3. Во второй уровень нужно добавить 3 узла и назвать их критериям из условия (сложность контроля, жаль н/с-цу, удаленность).

20. Повторяем действия, описанные в пунктах 11-15.

21. Сохраняем проект выполняя команду “Файл” – “Сохранить”. В результате получаем схему, изображенную на рис. 3:

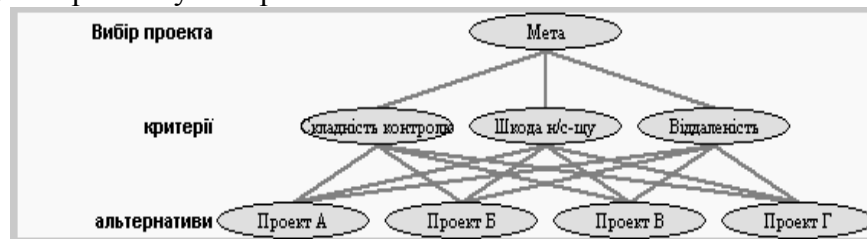


Рис. 3. Схема работы СППР (иерархия потерь)

Для проверки правильности проделанной работы нужно запустить процесс расчета командой “Проект” – “Расчет”, если все сделано верно, начнется процесс подсчета. В противном случае нужно исправить ошибки. Самая распространенная ошибка-недостаточность связей между узлами, что легко исправить добавив их или заменив другими.

Задания по вариантам:

1. Создать проект выбора мобильных телефонов.
2. Создать проект выбора телевизоров.
3. Создать проект выбора холодильников.
4. Создать проект выбора процессоров.
5. Создать проект выбора ОЗУ.
6. Создать проект выбора Винчестеров.
7. Создать проект выбора колонок.
8. Создать проект выбора магнитол.

9. Создать проект выбора приводов.
10. Создать проект выбора мониторов.
11. Создать проект выбора мышек.
12. Создать проект выбора клавиатур.
13. Создать проект выбора материнских плат.
14. Создать проект выбора принтеров.
15. Создать проект выбора модемов.
16. Создать проект выбора ксероксов.
17. Создать проект выбора стационарных телефонов.
18. Создать проект выбора охлаждающих систем (имеется в виду кулеры или водяное охлаждение).
19. Создать проект выбора тюнеров.
20. Создать проект выбора бесперебойников.

Лабораторная работа 5. Исследование с помощью системы "Выбор"

Тема: Работа с СППР "Выбор", средством создания проектов типа "Стоимость-эффективность". Расчеты, представление информации, выводы

Цель занятия:

1. Изучение возможностей проекта типа "Стоимость-эффективность" в программе "Выбор" как элемента моделирования СППР, завершение проекта предыдущего практического занятия

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения:

1. Отличие матрицы прямых сравнений в проектах "Проблема выбора" и "Стоимость-эффективность".

2. Дайте общую характеристику задачи "Стоимость-эффективность".

3. Перечислите классы задач, которые может решать проект "Стоимость-эффективность".

4. Преимущества и недостатки проекта "Стоимость-эффективность".

Ход работы

1. Изучить основные аспекты вычислений в СППР "Выбор".

2. Ознакомиться с интерфейсом и основными командами для проведения расчетов в СППР "Выбор".

3. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.

4. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР “Выбор” в СППР, описание примеров, иллюстративный материал.

Практическое задание.

Произвести нужные вычисления и оформить выводы с предварительно сделанным проектом выбора проекта.

Ход выполнения работы:

1. Предварительный проект командой “Файл” – “Открыть”
2. Командой “Проект” – “Расчет”, вызываем окно, изображенное на рис. 4:

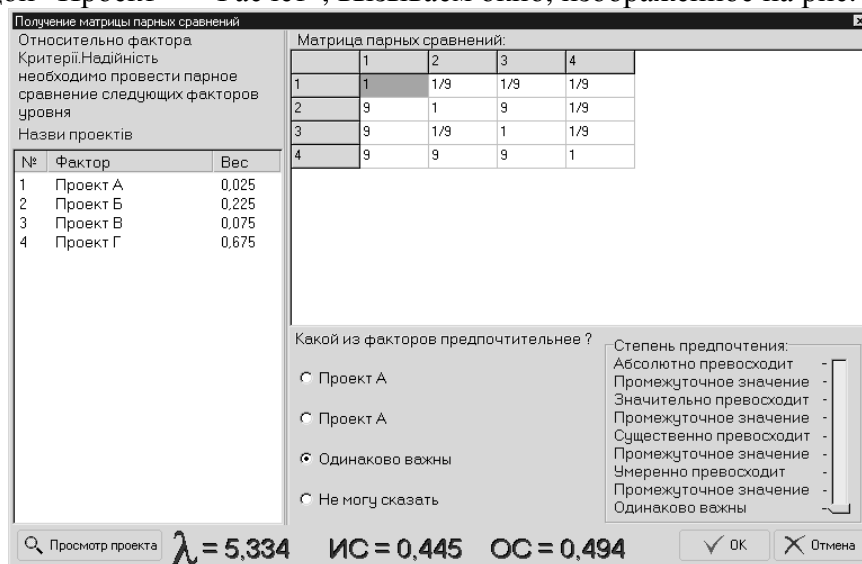


Рис. 4. Окно “Получение матрицы парных сравнений”

3. Данное окно представляет собой нечто вроде тестирующей программы оператором которого является матрица, которую нужно заполнить значениями. Для этого нужно нажимать на каждой ячейке под главной диагональю и отвечать на вопросы. Когда каждое поле будет заполнено нужно нажать ОК программа обработает информацию и выведет следующее окно (рис. 5):

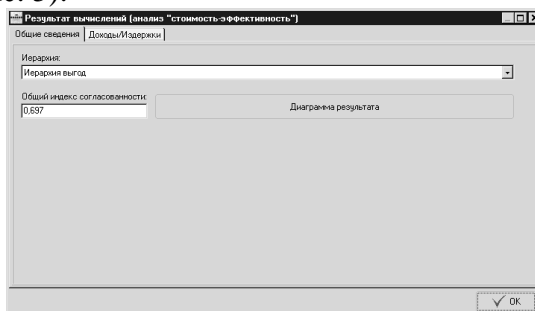


Рис. 5. Окно “Результат вычислений (анализ “стоимость-эффективность”)”.

4. Индекс согласованности показывает на то что данные которые Вы ввели не противоречивыми.

5. Далее нужно вывести диаграмму результата нажав на аналогичную клавишу в окне “Результат Вычислений”. Будет выведена диаграмма иерархии предпочтений (рис. 6). Для вывода иерархии потерь необходимо вернуться в окно “Результат вычислений (анализ “стоимость-эффективность”)” и в поле “Иерархия” выбрать параметр “Иерархия издержек”.

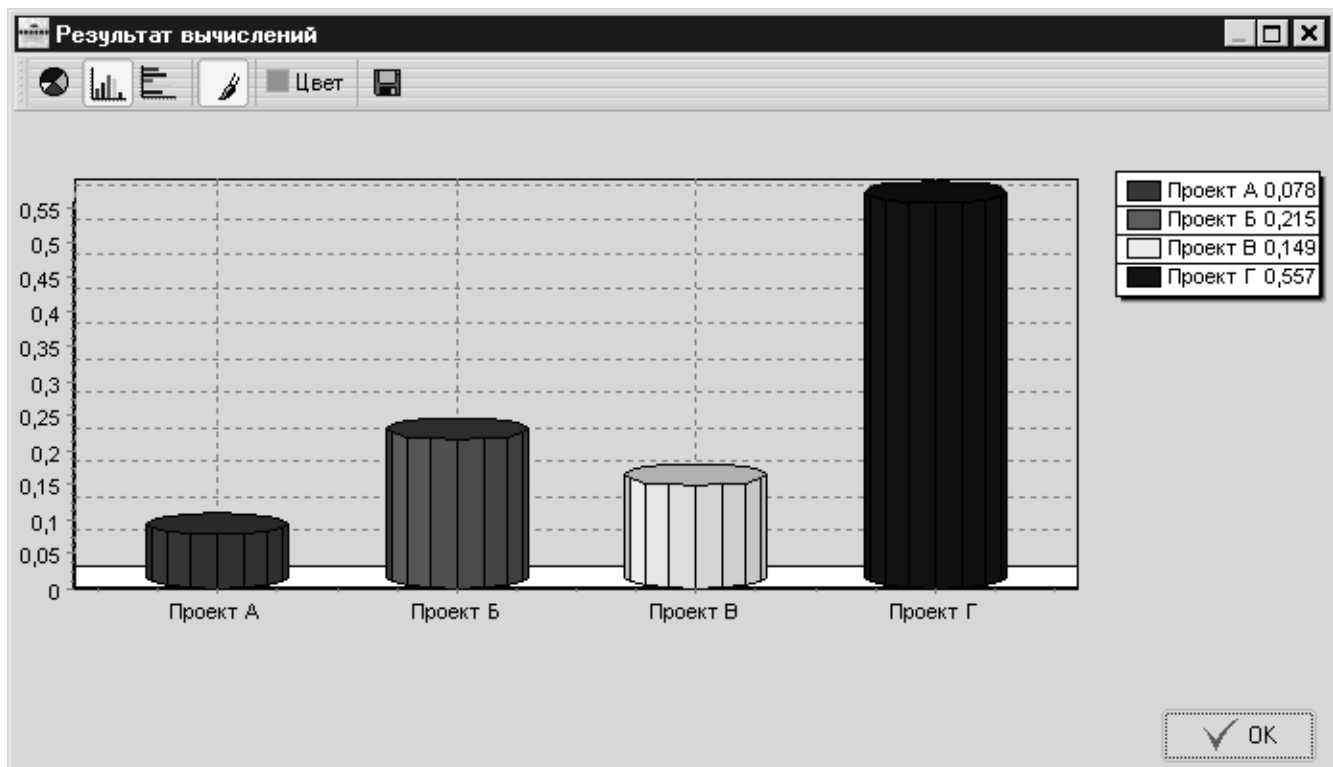


Рис. 6. Окно “Результат вычислений”

6. В данном окне можно выбрать вид диаграммы, цвет, можно также сохранить ее. Нужно сохранить проект. Диаграмма и расчеты сохраняются вместе с проектом. Задание к лабораторной работе брать с предыдущей работы.

Лабораторная работа 6. Исследование с помощью системы "Выбор"

Тема: Работа с СППР “Выбор”. Расчеты, представление информации, выводы

Цель занятия:

1. Изучение возможностей проекта типа “Поиск наилучшего решения” в программе “Выбор” как элемента моделирования СППР.

2. Формирование компетенций:

УК-2: Способен управлять проектом на всех этапах его жизненного цикла;

УК-2.3: Объясняет цели и формулирует задачи, связанные с подготовкой и реализацией проекта, управляет проектом на всех этапах его жизненного цикла.

ОПК_4: Способен применять на практике новые научные принципы и методы исследований

ОПК-4.2: Применяет на практике новые методы исследований

ОПК-7: Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

ОПК-7/2: Разрабатывает и применяет математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений

Вопросы для обсуждения

1. Теория важности критериев.
2. Свёртка критериев. Однородность критериев.

3. Методы определения качественной важности критериев.
4. Определение количественной важности критериев.
5. Методы определения коэффициентов важности критериев.

Ход работы

1. Изучить основные аспекты вычислений в СППР «Выбор».
2. Ознакомиться с интерфейсом и основными командами для проведения расчетов в СППР «Выбор».
3. Выполнить дополнительное задание лабораторной работы и дать ответы на дополнительные вопросы.
4. Оформить отчет по лабораторной работе, который включает выводы относительно возможностей использования средств анализа данных в СППР «Выбор» в СППР, описание примеров, иллюстративный материал.

С помощью программы «Выбор» попытаемся решить следующую задачу. Нам необходимо произвести отбор кандидатов на освободившуюся должность заместителя начальника отдела информатизации из числа сотрудников отдела. Кандидатов будем оценивать по нескольким критериям:

- стаж работы в организации,
- ответственность,
- образование,
- коммуникабельность.

Мы имеем 4-х претендентов на эту должность.

Таблица 1 – Критерий кандидатов

Ф.И.О.	Критерий			
	Стаж работы	Ответственность	Коммуникабельность	Образование
Иванов	5	Очень ответственный	Коммуникабельный	Высшее техническое
Петров	2	Достаточно ответственный	Замкнут (не коммуникабельный)	Высшее гуманитарное
Сидоров	1	Не ответственный	Очень коммуникабельный	Средне специальное
Потапов	3	ответственный	Достаточно коммуникабельный	Незаконченное высшее

С помощью программы попытаемся проанализировать, кто из претендентов наиболее подходит.

Сначала нам необходимо ввести в программу данные по критериям и фамилии претендентов. (Рис. 3)

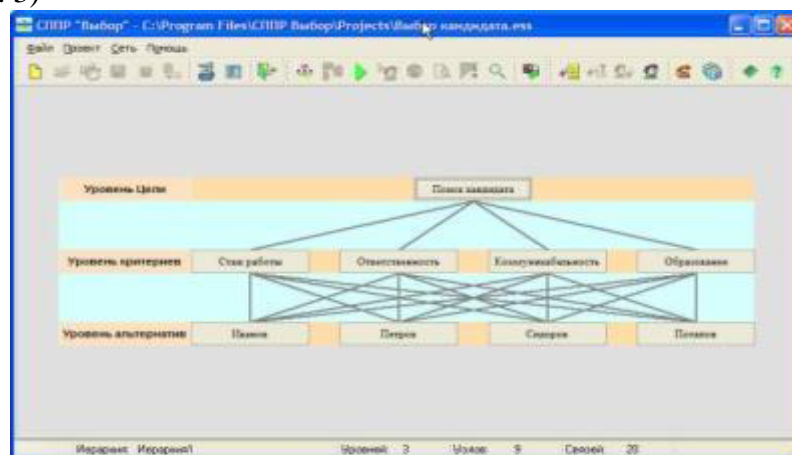


Рисунок 3 – Данные по критериям

Затем мы запускаем выполнение вычислений, где нам необходимо относительно каждого уровня произвести оценку нескольких факторов, тем самым расставив предпочтения (рис. 4)

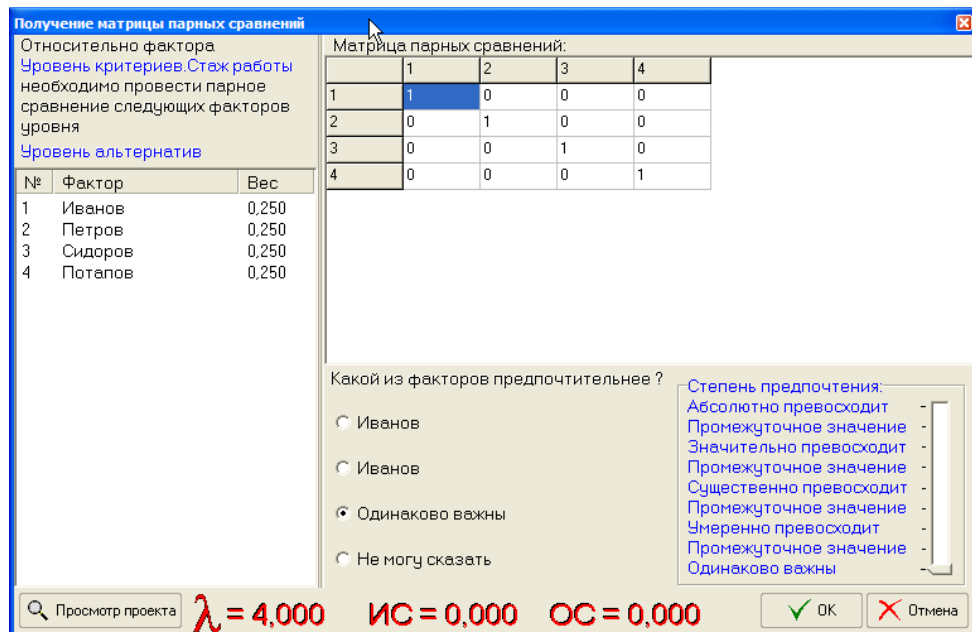


Рисунок 4 – Расчет предпочтений

Для каждого критерия производим оценку, допустим по стажу работы Иванов имеет большее предпочтение так как он проработал на нашем предприятии дольше Петрова. (рис. 5)

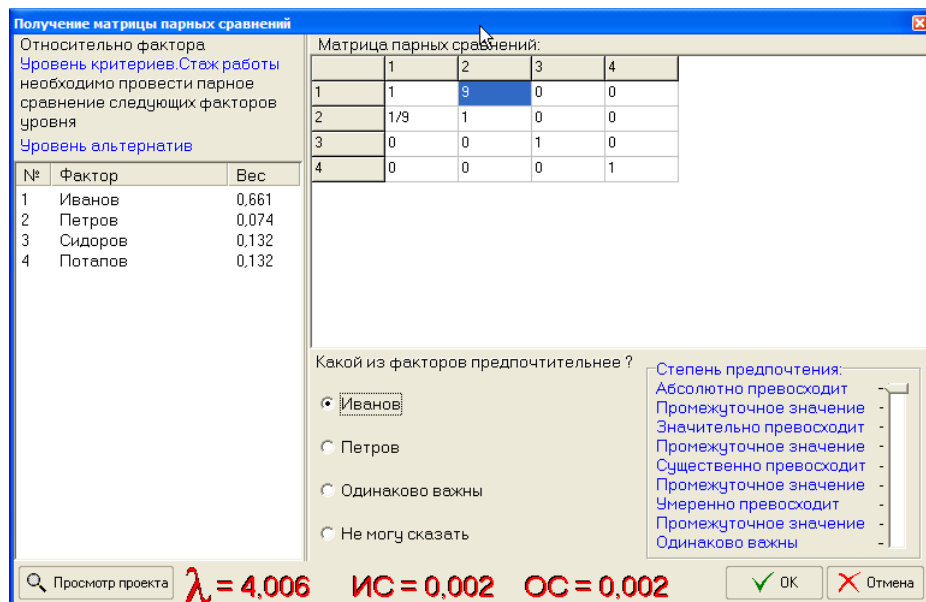


Рисунок 5 – Оценка критериев

Полученные матрицы парных сравнений по критерию стаж работы (рис. 6):

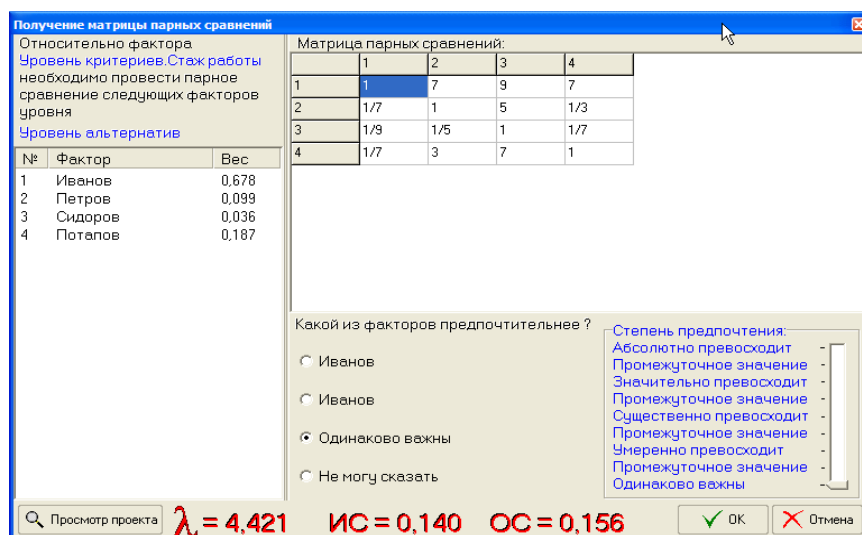


Рисунок 6 – Сравнение по критерию стажа

Иванов проработал дольше всех на предприятии, поэтому ему достается самая высокая оценка, а Сидоров проработал мало, у него самая маленькая.

Полученные матрицы парных сравнений по критерию ответственность (рис. 7):

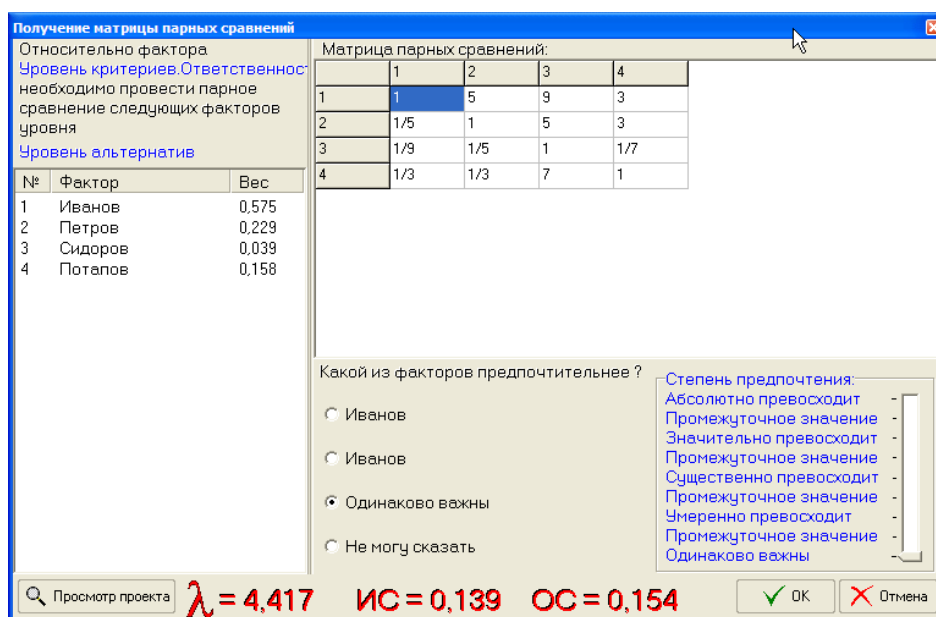


Рисунок 7 – Сравнение по критерию ответственности

Из этой матрицы мы видим, что наибольшая оценка опять достается Иванову, он признан самым ответственным всегда выполняя поручения, Сидоров же наоборот очень часто срывает сроки и относился халатно, поэтому у него самая маленькая оценка.

Полученные матрицы парных сравнений по критерию коммуникабельность (рис. 8):

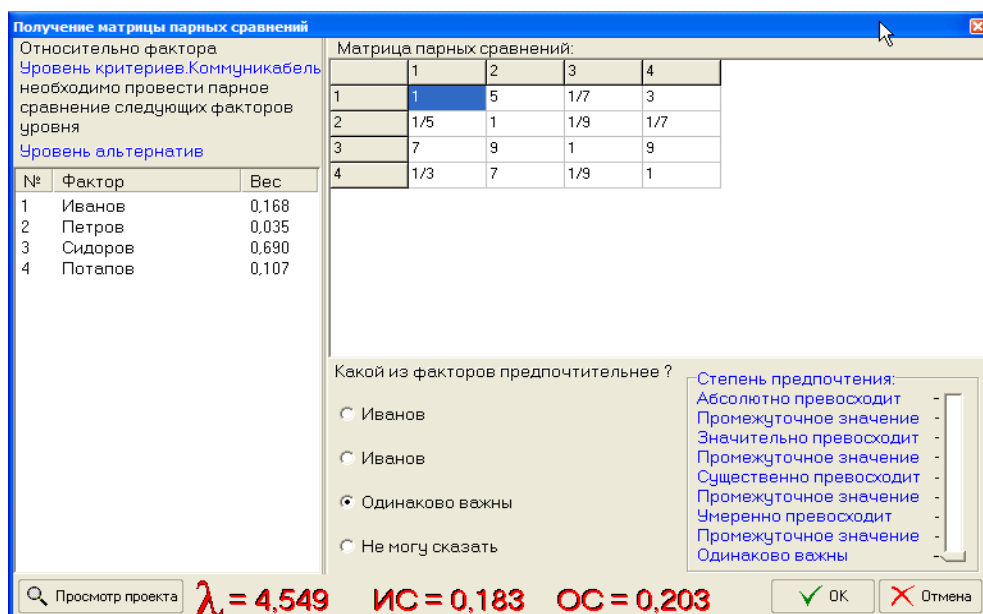


Рисунок 8 – сравнение по критерию коммуникабельности

Здесь мы видим очень интересную картину, по нашим наблюдениям наиболее коммуникабельным признан Сидоров, который по предыдущим критериям был аутсайдером, а вот Петров оказался позади всех, и признан замкнутым человеком.

Комментирую эту матрицу мы должны вспомнить постановку задачи, нам нужен заместитель начальника отдела информатизации, т.е. это должен быть человек хорошо разбирающийся в работе отдела, а значит техник по образованию, но с другой стороны это руководящая должность и возможно человек долго проработавший в отделе хоть и с гуманитарным образованием должен иметь равные шансы. Поэтому как мы видим на рис. 9, Иванову и Петрову проставлены одинаковые оценки, меньше всего у Сидорова т.к. руководитель должен иметь высшее образование.

Полученные матрицы парных сравнений по критерию образование (рис. 9):

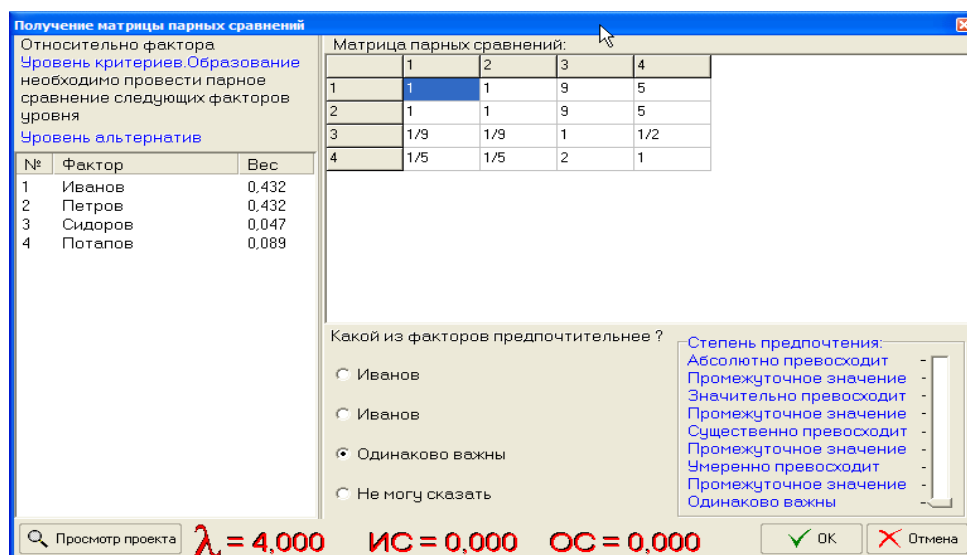


Рисунок 9 – Сравнение по критерию образования

Что для нас является важнее? Стаж, образование, коммуникабельность или ответственность? Думаю, уровнем шансы, нам нужен кандидат в равной степени удовлетворяющий всем параметрам. Расставляем приоритеты поровну. (рис. 10)

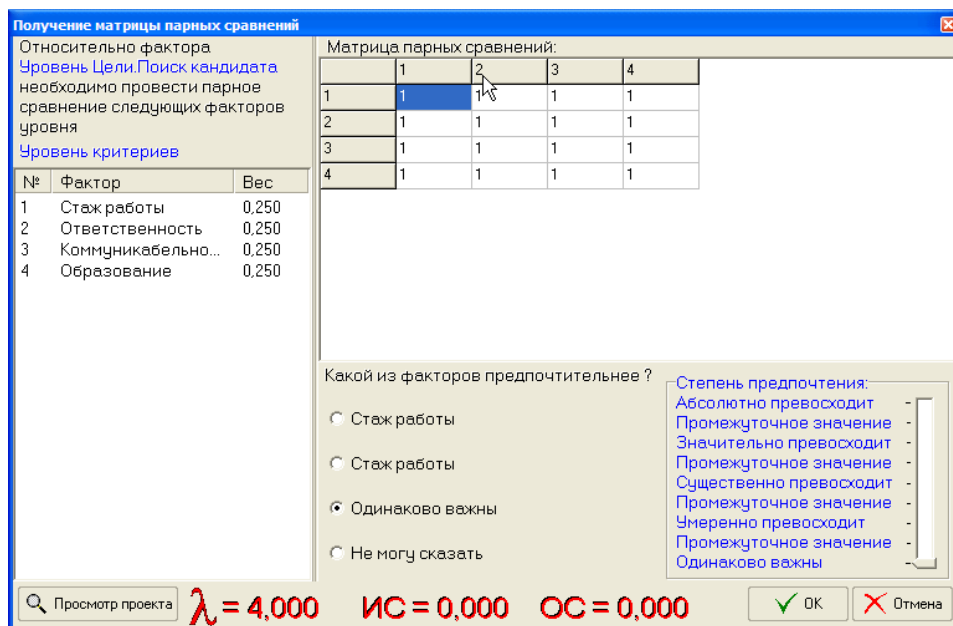


Рисунок 10 – Общие критерий кандидатов

Вычисления закончены, получаем результат (рис. 11):

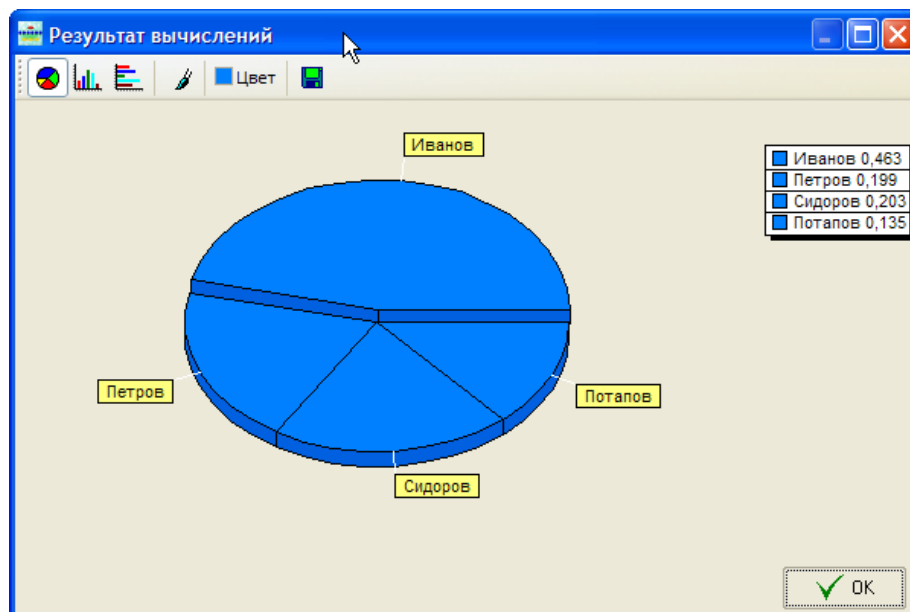


Рисунок 11-Получение результатов

Очевидно, что наиболее подходящим кандидатом на должность заместителя начальника отдела является Иванов.

Задание 1

С помощью программы «Выбор» решить следующую задачу. Нам необходимо произвести отбор кандидатов на освободившуюся должность старосты из числа студентов вашей группы. Кандидатов оценивать по нескольким критериям:

- успеваемость,
- ответственность,
- образование,
- коммуникабельность.

Произвести нужные вычисления и оформить выводы с предварительно сделанным проектом.

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

6.1.1. Основная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л1.1	Петров, А. Е.	Математические модели принятия решений: учебно-методическое пособие	Москва: Издательский Дом МИСиС, 2018	http://www.iprbookshop.ru/78572.html
Л1.2	Муромцев, Д. Ю., Шамкин, В. Н.	Методы оптимизации и принятие проектных решений: учебное пособие для магистрантов по направлению 11.04.03	Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2015	http://www.iprbookshop.ru/63866.html
Л1.3	Горелик, В. А.	Теория принятия решений: учебное пособие для магистрантов	Москва: Московский педагогический государственный университет, 2016	http://www.iprbookshop.ru/72518.html
6.1.2. Дополнительная литература				
	Авторы,	Заглавие	Издательство, год	Адрес
Л2.1	Бережная, О. В., Бережная, Е. В.	Методы принятия управленческих решений: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2015	http://www.iprbookshop.ru/62960.html
	Авторы,	Заглавие	Издательство, год	Адрес
Л2.2	Казанская, О. В., Юн, С. Г., Альсова, О. К.	Модели и методы оптимизации. Практикум: учебное пособие	Новосибирск: Новосибирский государственный технический университет, 2012	http://www.iprbookshop.ru/45397.html
6.1.3. Методические разработки				
	Авторы,	Заглавие	Издательство, год	Адрес
Л3.1	Палинчак, Н. Ф., Ярославцева, В. Я.	Системный анализ, оптимизация и принятие решений: методические указания и задания для самостоятельной работы	Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2014	http://www.iprbookshop.ru/55156.html
Л3.2	Артюхин Г. А.	Теория систем и системный анализ. Практикум принятия решений: Учебное пособие	Казань: Казанский государственный архитектурно-строительный университет, ЭБС АСВ, 2016	http://www.iprbookshop.ru/73321.html



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе
(ТИС (филиал) ДГТУ в г.Ставрополе)**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Математические модели информационных
процессов» для студентов направления подготовки
09.04.02 Информационные системы и технологии
Направленность (профиль) Информационные системы и
технологии

Методические указания по дисциплине «Математические модели информационных процессов» содержат задания для студентов, необходимые для лабораторных занятий.

Проработка предложенных заданий позволит студентам приобрести необходимые знания в области изучаемой дисциплины.

Предназначены для студентов направления подготовки 09.04.02 Информационные системы и технологии, направленность (профиль) Информационные системы и технологии

Содержание

Введение

Лабораторная работа 1 Исследование модели непрерывно-детерминированной системы

Лабораторная работа 2 Исследование стохастической имитационной модели системы

Лабораторная работа 3 Исследование элементов модели системы массового обслуживания

Лабораторная работа 4 Исследование модели системы массового обслуживания

ВВЕДЕНИЕ

При изучении курса наряду с овладением студентами теоретическими положениями уделяется внимание приобретению практических навыков, с тем, чтобы они смогли успешно применять их в своей последующей работе.

Цель освоения дисциплины – освоение методов разработки математических моделей информационных процессов и методологии и технологии математического моделирования при исследовании, проектировании, эксплуатации информационных систем; формирование общекультурных и профессиональных компетенций магистра в соответствии с требованиями ФГОС по направлению Информационные системы и технологии; подготовка магистра к деятельности, требующей применение научно-практических знаний и умений в области анализа информационных процессов; развитие логического, алгоритмического мышления студентов, умения самостоятельно расширять свои знания в области математического представления информационных процессов.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

В результате освоения данной дисциплины формируется следующая компетенция у обучающегося:

ОПК-6.2: Оценивает процессы получения, передачи, хранения и представления информации на основе положений системной инженерии.

Изучив данный курс, студент должен:

Знать:

Инструментальные средства математического моделирования информационных процессов и условия их применимости.

Теоретические основы функционирования информационных систем и процессов и математический аппарат для их описания.

Уметь:

Создавать и исследовать математические модели информационных процессов с использованием стандартных пакетов автоматизированного проектирования.

Владеть:

Созданием математических моделей информационного процесса на языке высокого уровня и с использованием интегрированных пакетов прикладных программ.

Оценки адекватности математической модели.

Реализация компетентного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Лекционный курс является базой для последующего получения обучающимися практических навыков, которые приобретаются на практических занятиях, проводимых в активных формах: деловые игры; ситуационные семинары. Методика проведения практических занятий и их содержание продиктованы стремлением как можно эффективнее развивать у студентов мышление и интуицию, необходимые современному специалисту. Активные формы семинаров открывают большие возможности для проверки усвоения теоретического и практического материала.

Лабораторная работа 1 Исследование модели непрерывно-детерминированной системы

Целью занятия является освоение методов моделирования непрерывно-детерминированных динамических систем. В процессе выполнения задания студенты должны:

разработать модели заданной системы, реализованные в пакете визуального моделирования Simulink на основе полученного описания;

получить результаты исследования данных моделей и выполнить их объяснение в терминах прикладной области.

В качестве примера учебного задания рассматривается задача об управлении протеканием воды через систему, состоящую из двух цилиндрических баков, расположенных на разной высоте, или, более кратко, задача о двух баках. Выбор этой задачи в качестве примера обусловлен следующими причинами:

понятным физическим принципом;

нелинейным поведением компонент системы;

наличием аварийных ситуаций в поведении системы, которые необходимо

обрабатывать;

Модель системы, реализованная в подсистеме Simulink пакета Matlab

Построение модели в Simulink сводится к перемещению с помощью мыши необходимых блоков из библиотек Simulink в окно создаваемой модели и соединению этих блоков между собой с помощью функциональных связей. Библиотеки Simulink содержат большое количество разнообразных функциональных блоков, которые отображаются на экране в виде пиктограмм.

Используя описание задачи о двух баках можно выделить в данном примере подсистему System (рис.3), представляющую собой составной блок, который содержит в себе функциональную схему, содержащую в себе диаграмму Stateflow (представленную блоком Controller, являющимся экземпляром стандартного блока Chart и описывающую поведение контроллера) и составной блок Tank_System_Block, соединенные соответствующими функциональными связями (рис.4). Так же в блоке System присутствуют часы Clock, подающие системное время из Simulink в диаграмму Stateflow. Это объясняется тем, что при построении моделей, где используются блоки Stateflow (содержащие в себе переходы, инициируемые истечением неких временных интервалов) необходимо синхронизировать внутреннее системное время в Stateflow и в Simulink. Для этого необходимо на отдельный вход в блоке Stateflow подавать системное время из Simulink и именно это время использовать при составлении условий переходов (таких как истечение отрезков времени Time1 и Time2 в задаче о двух баках). Как видно из рис.4, к связи, соединяющей блок Clock и Controller, подсоединены два экземпляра блока Hit Cross. Еще два экземпляра блока Hit Cross подсоединены к связи блока Controller и выхода h2 блока Tank_System_Block. Использование Hit Cross блоков необходимо для правильного выполнения переходов в диаграмме Stateflow, включенной в модель

Simulink, в которой происходит непрерывное интегрирование. На вход Hit Cross блока подается некая величина (в модели двух баков для двух Hit Cross блоков это системное время, для еще двух - величина h2). Сам Hit Cross блок содержит в себе некую величину, при совпадении которой с величиной, подаваемой на вход Hit Cross блока, система уменьшает шаг интегрирования. В связи со спецификой системы Stateflow это влияет на правильное выполнение переходов и не позволяет системе «проскочить момент», когда должен выполняться тот или иной переход. У первого Hit Cross блока внутренней величиной является момент времени Time1, у второго - Time 1+Time2, у третьего - значение L_plus, у четвертого - L_minus.

Блок System не имеет входов и имеет два выхода - h1 и h2, соединенные со стандартным блоком Mux (объединяющим их в вектор (h1,h2)), соединенный со стандартным блоком вывода Scope (на который он подает вектор выходных величин (h1,h2)).

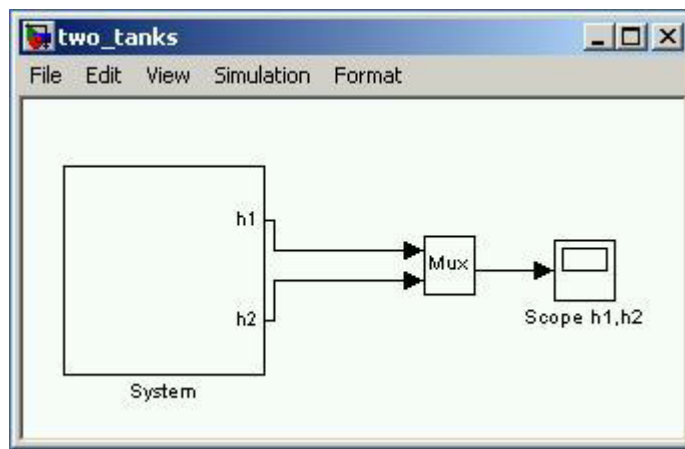


Рисунок 3

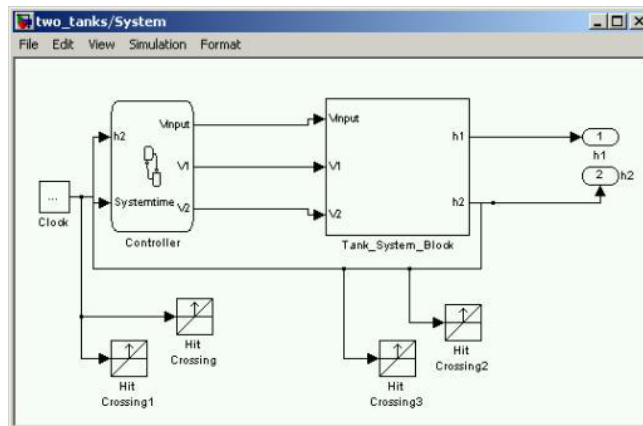


Рисунок 4

В свою очередь, составной блок Tank_System_Block содержит в себе функциональную схему, состоящую из составного блока Tank_System (имеющего три входа, соединенные с соответствующими выходами блока Chart, на которые подаются сигналы контроллера, и два выхода - h1 и h2) и пар экземпляров стандартных блоков Const, Relational Operator и Stop Simulation (выполняющих проверку на переполнение баков), соединенных соответствующими функциональными связями (рис.5). Блок Tank_System_Block имеет три входа (на которые подаются сигналы от контроллера) и два выхода (h1 и h2).

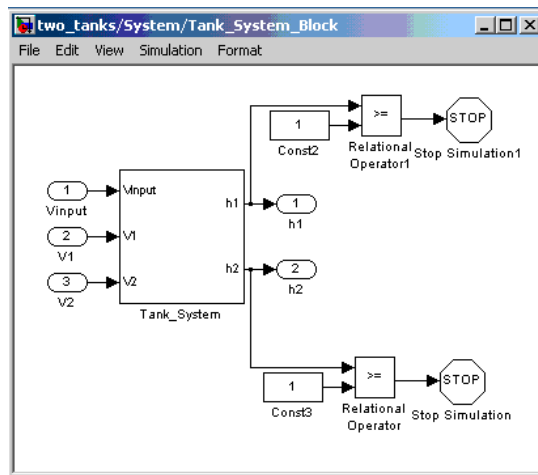


Рисунок 5

Составной блок Tank_System содержит в себе функциональную схему, состоящую из составных блоков Two_Tanks, Vin_Control, K1_Control и K2_Control, соединенных между собой функциональными связями (рис.6). Блок Tank_System имеет то же количество входов и выходов с теми же значениями, что и предыдущий блок.

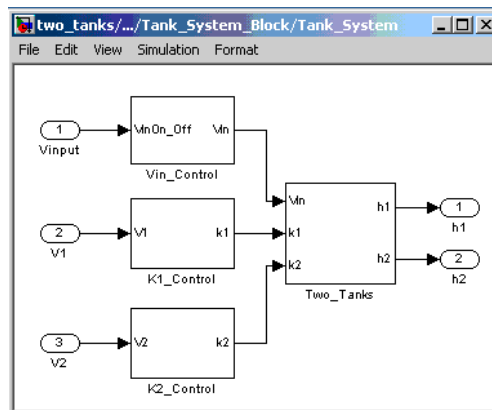


Рисунок 6

Составной блок Vin_Control содержит в себе функциональную схему, состоящую из стандартного блока Switch и двух экземпляров стандартного блока Const, содержащих в себе значения, соответствующие скорости входного потока воды в систему двух баков, когда входной кран открыт и закрыт. Переключения между этими значениями происходит в блоке Switch в зависимости от значения, поступающего на вход блока Vin_Control от контроллера. Соответствующее значение Vin подается на выход. Блок Vin_Control имеет один вход и один выход (рис.7):

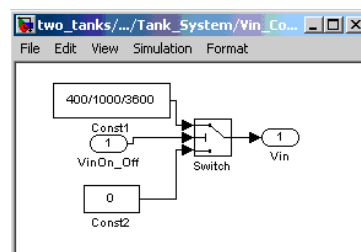


Рисунок 7

Составные блоки K1_Control и K2_Control содержат в себе идентичные функциональные схемы, различающиеся только значением коэффициентов в уравнении, представленном в стандартном блоке fcn. Функциональная схема состоит из двух экземпляров стандартного блока Switch (один из которых необходим для переключения между положениями кранов V1/V2 открыто/закрыто, а второй для отслеживания ситуации, когда параметр p становится равным 80 и соответствующего переключения между значениями функции K(p)), трех экземпляров стандартного блока Const (содержащих в себе значения, 1 и -1, соответствующие положению крана V1/V2 (открывается/закрывается) и значение 0 для функции K(p) в ситуации, когда $p \geq 80$), стандартного блока Integrator (интегрирующего поступающее от переключателя значение в пределах от 0 до 80 с начальным значением интегрируемой величины 80) и стандартного блока f(u), в котором происходит вычисление значения функции K1(p)/K2(p). Переключения в блоке Switch происходит в зависимости от значения, поступающего на вход блока K1_Control/K2_Control от контроллера. Соответствующее значение k1/k2 подается на выход. Блок K1_Control/K2_Control имеет один вход и один выход (на рис.8 представлен блок K1_Control, блок K2_Control идентичен):

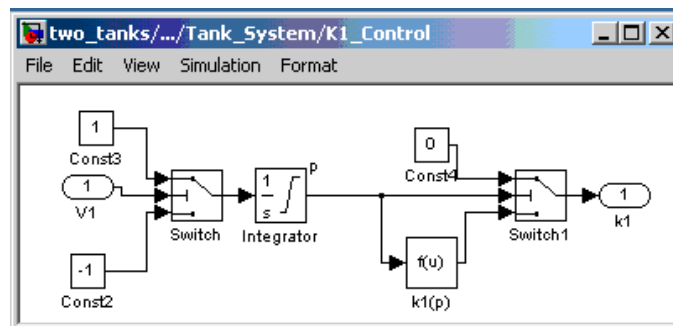


Рисунок 8

Составной блок Two_Tanks содержит в себе функциональную схему, состоящую из составных блоков Vout_calc, V12_calc1 и V12_calc2 (отвечающих за вычисление значений Vout и V12), стандартного блока Switch, двух экземпляров стандартного блока Mux (объединяющих в вектора соответствующие значения - Vin (поступающее на вход блока Two_Tanks с выхода блока Vin_Control)) и V12, а также V12 и Vout), двух экземпляров стандартного блока fcn (производящих вычисление производных h1 и h2), двух экземпляров стандартного блока Integrator, выходами которых являются значения h1 и h2, поступающие на выход. Переключения в блоке Switch происходит в зависимости от значения h2, поступающего на вход блока Switch от интегратора, и в зависимости от этого значения, подключается один из блоков, вычисляющих V12. Блок Two_Tanks имеет три входа, на которые подаются значения Vin, k1 и k2; и два выхода - h1 и h2 (рис.9):

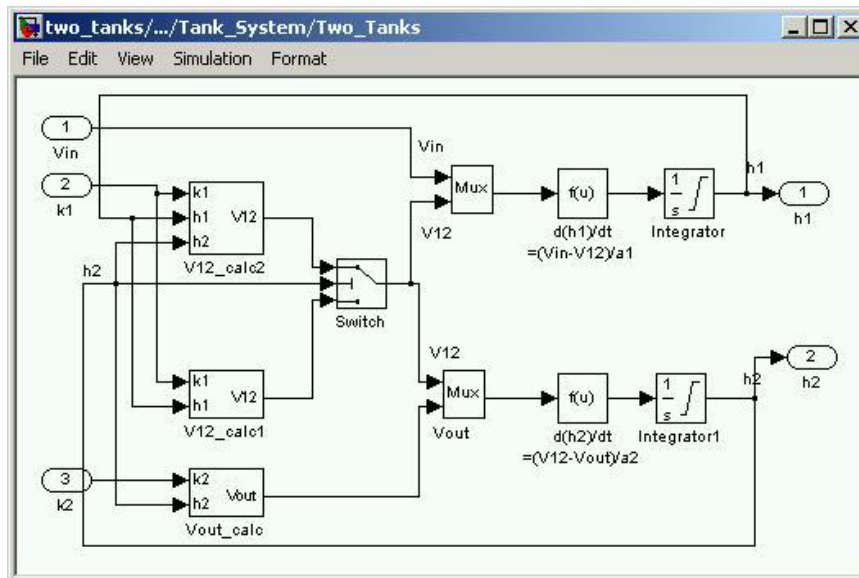


Рисунок 9

Составной блок Vout_calc содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k2 и h2, поступающие на вход блока Vout_calc, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины Vout. Блок Vout_calc имеет два входа и один выход (рис.10):

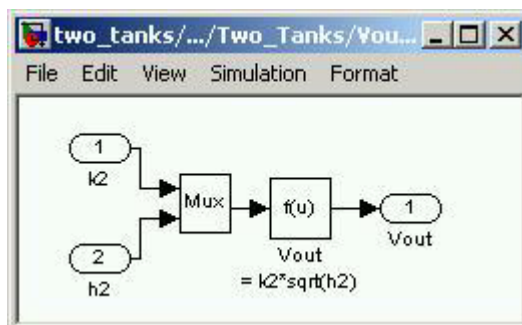


Рисунок 10

Составной блок V12_calc1 содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k1 и h1, поступающие на вход блока V12_calc1, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины V12. Блок V12_calc1 имеет два входа и один выход (рис.11):

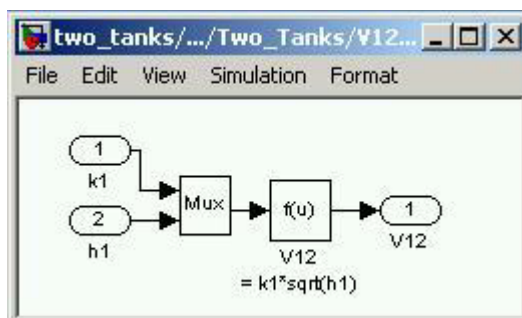


Рис. 11

Составной блок V12_calc2 содержит в себе функциональную схему, состоящую из стандартного блока Mux, объединяющего в один вектор значения k1, h1 и h2, поступающие на вход блока V12_calc2, и подающего этот вектор на вход стандартного блока fcn, в котором происходит вычисление выходной величины V12. Блок V12_calc2 имеет три входа и один выход (рис.12):

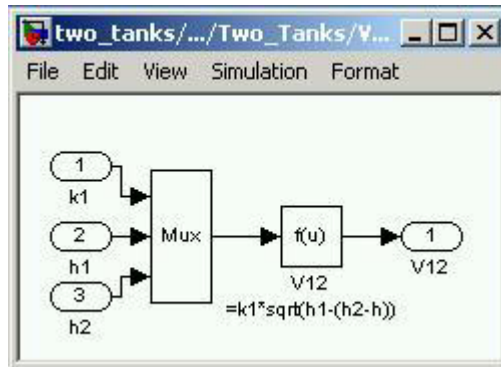


Рис. 12

Диаграмма Stateflow, представлена экземпляром стандартного блока Chart, (рис.13). В ней вместо внутреннего времени t используется подаваемое на вход блока системное время Simulink (входная переменная Systemtime).

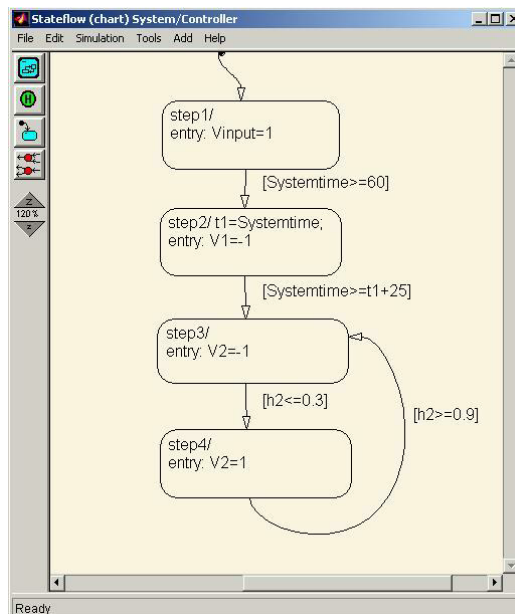


Рис. 13

Результаты эксперимента

На рис.14 представлена временная диаграмма изменений уровней воды в баках, полученная в результате эксперимента с моделью, реализованной в подсистеме Simulink пакета Matlab:

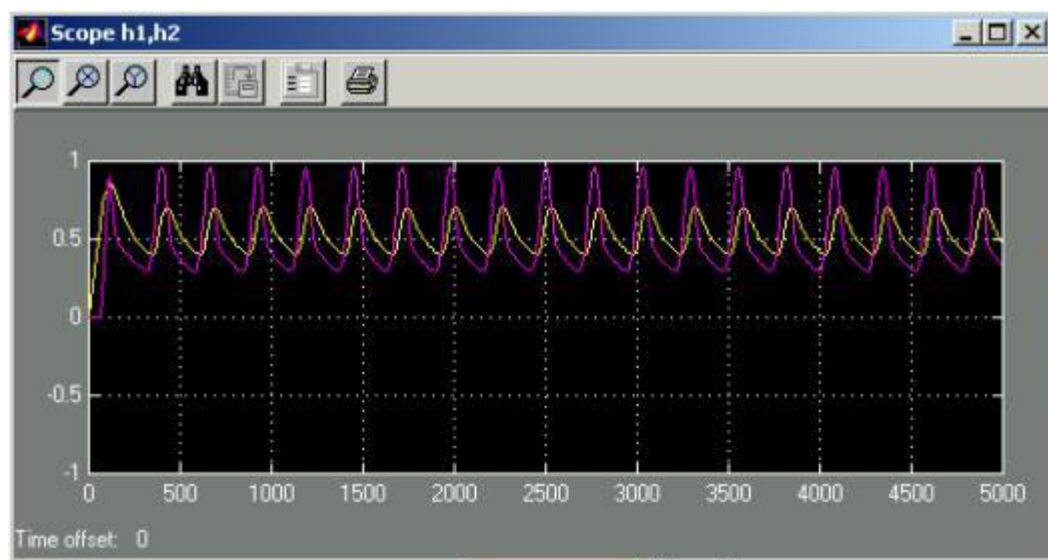


Рис. 14

Данные временные диаграммы отражают изменения уровней воды в системе двух баков при открытии/закрытии выходного крана. Возрастающие участки кривых $h_1(t)$ и $h_2(t)$ соответствуют состоянию системы, когда выходной кран закрыт. Убывающие участки кривых $h_1(t)$ и $h_2(t)$ соответствуют состоянию системы, когда выходной кран открыт.

Лабораторная работа №2 Исследование стохастической имитационной модели системы

Целью занятия является практическое освоение методов моделирования стохастических динамических систем. В процессе выполнения задания студенты должны:

разработать модель заданной системы с использованием пакета визуального моделирования Simulink на основе полученного описания;

получить результаты исследования модели и выполнить их объяснение в терминах прикладной области.

1. Описание моделируемой системы

В практике передачи и приема информации с использованием радиоканалов связи одной из важнейших задач является оценка влияния помех на вероятность безошибочной передачи информации. Рассмотрим систему цифрового радиоканала, представленную на рис. 1.

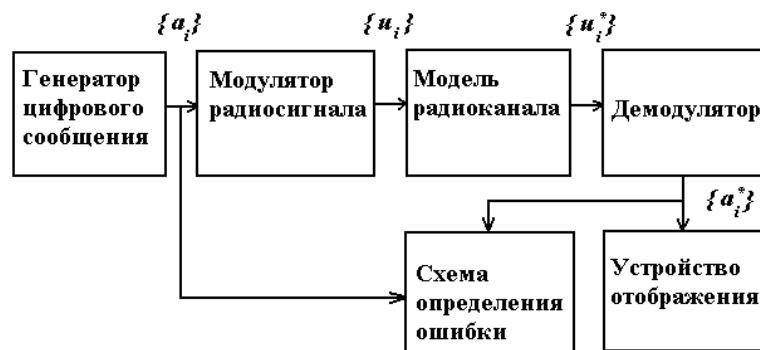


Рисунок 1

Генератор цифровых сообщений служит для имитации двоичных кодов заданной длины и может быть представлен следующим выражением

$$\begin{cases} a_i = 0, \text{ если } Rnd(i) < 0,5 \\ a_i = 1, \text{ если } Rnd(i) \geq 0,5 \end{cases}$$

где $Rnd(i)$ число из интервала $0 \dots 1$, выдаваемое встроенным программным генератором псевдослучайных чисел; $i = \overline{1, I_{зад}}$.

Несущий радиосигнал задан выражением

$$u(t) = U \sin(2\pi f_n t + \varphi_0),$$

где U амплитуда радиосигнала; f_n - несущая частота; φ_0 - начальная фаза.

Модуляция радиосигнала по закону передаваемого сообщения может быть выполнена с помощью импульсной модуляции (ИМ):

$$\begin{cases} U = 0, \text{ если } a_i = 0; \\ U = 1, B, \text{ если } a_i = 1. \end{cases}$$

Частотной модуляции (ЧМ):

$$\begin{cases} f = f_n, \text{ если } a_i = 0; \\ f = f_n + \Delta f, \text{ если } a_i = 1. \end{cases}$$

Фазокодовой манипуляции (ФКМ):

$$\begin{cases} \varphi_0 = 0, \text{ если } a_i = 0; \\ \varphi_0 = \pi, \text{ если } a_i = 1. \end{cases}$$

Модель канала характеризуется видом помех. Для аддитивных помех, приводящих к искажению амплитуды сигнала, используется выражение

$$u^*(t) = u(t) \pm Rnd(t) \cdot U_{\Pi},$$

где U_{Π} - максимальное значение амплитуды помехи.

Модель канала с фазовыми искажениями имеет вид

$$u^*(t) = u(t) + U_{\Pi} \cdot \sin(2\pi f_n t \pm Rnd(t) \cdot \varphi_0).$$

Модель канала с преобладающими частотными искажениями может быть задана выражением

$$u^*(t) = u(t) + U_{\Pi} \cdot \sin(2\pi t(f_n \pm Rnd(t) \cdot \Delta f)).$$

Демодуляция определяется видом модуляции радиосигнала и для амплитудной импульсной модуляции задается выражением

$$\begin{cases} u^*(t) > U_{порога}^1, \text{ то } a_i^* = 1; \\ u^*(t) < U_{порога}^0, \text{ то } a_i^* = 0, \text{ иначе } a_i^* = x. \end{cases}$$

Пороговые значения нулевого и единичного уровней определяются исходя из условий распространения сигнала и дальности связи. Состояние x характеризует появление ошибки. Аналогичные выражения только с пороговыми значениями частоты или фазы сигнала используются для демодуляции ЧМ и ФКМ сигналов.

Фиксация ошибки осуществляется посредством сравнения $a_i = a_i^*$ или по наличию ошибки демодуляции $a_i^* = x$.

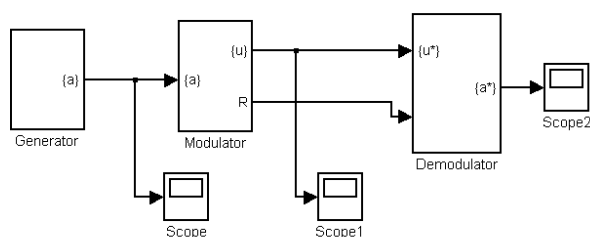
2. Задание для самостоятельной работы

В соответствии с номером по журнальному списку выбрать из табл. 1 исходные данные для моделирования и составить модель радиоканала. При составлении модели максимально использовать возможности пакета имитационного моделирования Simulink. Оформить отчет по практическому занятию в котором отразить текст описания системы и разработанной модели. Привести графики на выходе элементов модели для различных вариантов исходных данных. Оценить влияние различных помеховых воздействий на число ошибок.

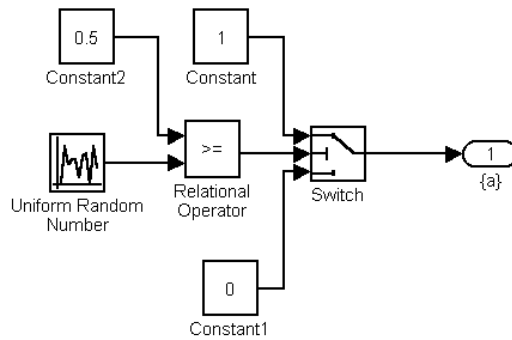
Таблица 1

№ п/п	Исходные данные для моделирования					Примечание
	Вид модуляции	Число разрядов	$U^1_{порога}$	$U^0_{порога}$	$U_{П}$	
1	ИМ	7	$0,6U$	$0,1U$	$0,01U$	
2	ФКМ	8				
3	ЧМ	9				$\Delta f = 0,1 f$
4	ИМ	10	$0,8U$	$0,2U$	$0,02U$	
5	ЧМ	7				$\Delta f = 0,2 f$
6	ФКМ	8				
7	ЧМ	9				$\Delta f = 0,01 f$
8	ИМ	10	$0,9U$	$0,1U$	$0,01U$	
9	ФКМ	7				
10	ИМ	8	$0,5U$	$0,05U$	$0,01U$	

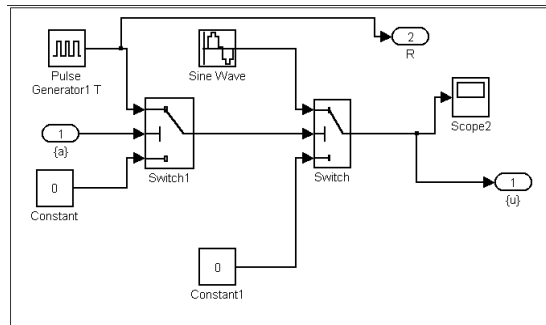
Структура модели



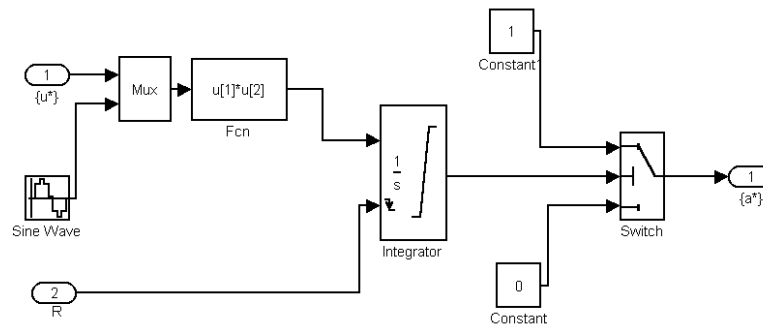
Модель генератора цифрового сигнала



Модель модулятора радиосигнала



Модель демодулятора радиосигнала



Исходные данные моделей

Время моделирования 1000.

Генератор случайных чисел:

Минимальное значение случайного числа =0; максимальное значение =1; начальное значение последовательности (любое целое число); Длительность одной реализации =100.

Minimum=0; Maximum=1; Initial seed=0; Sample time=100.

Генератор импульсов:

Период =100 секунд модели; Длительность импульса =50 (скважность равна 2); амплитуда 1 вольт; начальное время генерации =0.

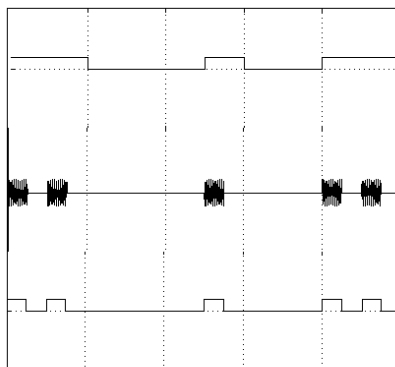
Period=100(secs); Duty cycle (% of period)=50; Amplitude=1; Start time=0.

Генератор синусоиды:

Амплитуда 1 вольт; Циклическая частота=10 радиан в секунду; Начальная фаза =0; Длительность одного значения 1.

Amplitude=1; Frequency (rad/sec)=10; Faze (rad)=0; Sample time=1.

Пример результатов моделирования



Лабораторная работа №3

Исследование элементов модели системы массового обслуживания

Цель занятия: Освоить методику моделирования элементов систем массового обслуживания (СМО)

1. Общие сведения о СМО

Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования.

В любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания заявкой и собственно обслуживание заявки. Это можно изобразить в виде некоторого i -го прибора обслуживания Π_i (рис. 1), состоящего из накопителя заявок H_i в котором может одновременно находиться $l_i = \overline{L_i^H}$ заявок, где L_i^H — емкость i -го накопителя, и канала обслуживания заявок (или просто канала) K_i . На каждый элемент прибора обслуживания Π_i поступают потоки событий: в накопитель H_i — поток заявок W_i на канал K_i — поток обслуживания u_i .

Потоком событий называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Различают потоки однородных и неоднородных событий. Поток событий называется *однородным*, если он характеризуется только моментами поступления этих событий (вызывающими моментами) и задается последовательностью $\{t_n\} = \{0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq \dots\}$, где t_n — момент наступления n -го события — неотрицательное вещественное число. Однородный поток событий также может быть задан в виде последовательности промежутков времени между n -м и $(n-1)$ -м событиями $\{\tau_n\}$, которая однозначно связана с последовательностью вызывающих моментов $\tau_n = t_n - t_{n-1}, n \geq 1, t_0 = 0$, т. е. $\tau_1 = t_1$.

Потоком неоднородных событий называется последовательность $\{t_n, f_n\}$, где t_n — вызывающие моменты; f_n — набор признаков события. Например, применительно к процессу обслуживания для неоднородного потока заявок могут быть заданы принадлежность к тому или иному источнику заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т. п.

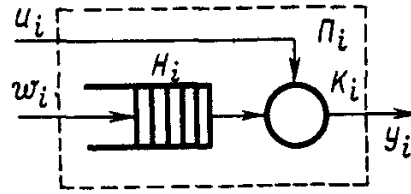


Рис. 1. Устройство обслуживания заявок

Рассмотрим поток, в котором события разделены интервалами времени τ_1, τ_2, \dots , которые вообще являются случайными величинами. Пусть интервалы τ_1, τ_2, \dots независимы между собой. Тогда поток событий называется *поток с ограниченным последствием*.

Пример потока событий приведен на рис. 2, где обозначено T_j интервал между событиями (случайная величина); T_n — время наблюдения, T_c — момент совершения события.

Интенсивность потока можно рассчитать экспериментально по формуле

$$\lambda = \frac{N}{T_n}$$

где N — число событий, произошедших за время наблюдения T_n . Если $T_j = \text{const}$ или определено какой-либо формулой $T_j = f(T_{j-1})$, то поток называется *детерминированным*. Иначе поток называется *случайным*.

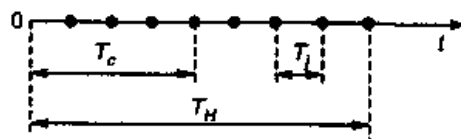


Рис.2 Графическое изображение N-схемы

Случайные потоки бывают:

- ординарными, когда вероятность одновременного появления 2-х и более событий равна нулю;
- стационарными, когда частота появления событий постоянная;
- без последствия, когда вероятность не зависит от момента предыдущих событий.

Поток событий называется *ординарным*, если вероятность того, что на интервал времени, примыкающий к моменту времени t , попадает больше одного события, пренебрежительно мала по сравнению с вероятностью того, что на этот же интервал времени попадает ровно одно событие.

Стационарным потоком событий называется поток, для которого вероятность появления того или иного числа событий на интервале времени t зависит от длины этого участка и не зависит от того, где на оси времени взят этот участок.

Интенсивность потока может быть любой неотрицательной функцией времени, имеющей размерность, обратную размерности времени. Для стационарного потока его интенсивность не зависит от времени и представляет собой постоянное значение, равное среднему числу событий, наступающих в единицу времени.

Возможные приложения. Обычно в приложениях при моделировании различных систем применительно к элементарному каналу обслуживания K_j , можно считать, что поток заявок $W_j \in W$, т. е. интервалы времени между моментами появления заявок (вызывающие моменты) на входе K_j образует подмножество неуправляемых переменных, а поток обслуживания $u_j \in U$, т. е. интервалы времени между началом и окончанием обслуживания заявки, образует подмножество управляемых переменных.

Заявки, обслуженные каналом K_j и заявки, покинувшие прибор Π_j по различным причинам необслуженными (например, из-за переполнения накопителя H_j , образуют выходной поток $y_j \in Y$, т. е. интервалы времени между моментами выхода заявок образуют подмножество выходных переменных.

Процесс функционирования прибора обслуживания Π_j можно представить как процесс изменения состояний его элементов во времени $z_j(t)$. Переход в новое состояние для Π_j означает изменение количества заявок, которые в нем находятся (в канале K_j , и в накопителе H_j). Таким образом, вектор состояний для Π_j имеет вид $\vec{z}(t) = (z_j^K, z_j^H)$, где z_j^H — состояние накопителя ($z_j^H = 0$ — накопитель пуст, $z_j^H = 1$ — в накопителе имеется одна заявка ... $z_j^H = L_j^H$ — накопитель полностью заполнен); L_j^H — емкость накопителя, измеряемая числом заявок, которые в нем могут поместиться; z_j^K — состояние канала ($z_j^K = 0$ — канал свободен, $z_j^K = 1$ — канал занят и т. д.).

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а *Q-схемы*, образуемые композицией многих элементарных приборов обслуживания (сети массового обслуживания). Если каналы K_i различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание (многоканальная *Q-схема*), а если приборы Π_j и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная *Q-схема*). Таким образом, для задания *Q-схемы* необходимо использовать оператор сопряжения R , отражающий взаимосвязь элементов структуры (каналов и накопителей) между собой.

Связи между элементами *Q-схемы* изображают в виде стрелок (линий потока,

отражающих направление движения заявок). Различают разомкнутые и замкнутые Q -схемы. В разомкнутой Q -схеме выходной поток обслуженных заявок не может снова поступить на какой-либо элемент, т. е. обратная связь отсутствует, а в замкнутых Q -схемах имеются обратные связи, по которым заявки двигаются в направлении, обратном движению вход-выход.

Собственными (внутренними) параметрами Q -схемы будут являться количество фаз обслуживания, количество каналов в каждой фазе, количество накопителей каждой фазы, емкость i -го накопителя. Следует отметить, что в теории массового обслуживания в зависимости от емкости накопителя применяют следующую терминологию для систем массового обслуживания: системы с потерями (т. е. накопитель отсутствует, а имеется только канал обслуживания), системы с ожиданием (т. е. накопитель, имеет бесконечную емкость и очередь заявок не ограничивается) и системы смешанного типа (с ограниченной емкостью накопителя). Всю совокупность собственных параметров Q -схемы обозначим как подмножество H ,

Для задания Q -схемы также необходимо описать алгоритмы ее функционирования, которые определяют набор правил поведения заявок в системе в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе и обслуживания заявок каналом, каждого элементарного обслуживающего прибора P , Q -схемы. Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов.

В зависимости от динамики приоритетов в Q -схемах различают статические и динамические приоритеты. Статические приоритета назначаются заранее и не зависят от состояний Q -схемы, т. е. они являются фиксированными в пределах решения конкретной задачи моделирования. Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций. Исходя из правил выбора заявок из накопителя на обслуживание каналом, можно выделить относительные и абсолютные приоритеты. *Относительный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель ожидает окончания обслуживания предшествующей заявки каналом K , и только после этого занимает канал. *Абсолютный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель прерывает обслуживание каналом K заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из K заявка может либо покинуть систему, либо может быть снова записана на какое-то место в H).

При рассмотрении алгоритмов функционирования приборов обслуживания (каналов, и накопителей) необходимо также задать набор правил, по которым заявки покидают P и K . Для P — либо правила переполнения, по которым заявки в зависимости от заполнения H покидают систему, либо правила ухода, связанные с истечением времени ожидания заявки в H , для K — правила выбора маршрутов или направлений ухода. Кроме того, для заявок необходимо задать правила, по которым они остаются в канале K или не допускаются до обслуживания каналом K т. е. правила блокировок канала. При этом различают блокировки K по выходу и по входу. Такие блокировки отражают наличие управляющих связей в Q -схеме, регулирующих поток заявок в зависимости от состояний Q -схемы. Весь набор возможных алгоритмов поведения заявок в Q -схеме можно представить в виде некоторого оператора алгоритмов поведения заявок A .

Таким образом, Q -схема, описывающая процесс функционирования системы массового

обслуживания любой сложности, однозначно задается в виде $Q = \langle W, U, H, Z, R, A \rangle$.

2. Моделирование основных элементов СМО с использованием пакета

Simulink

В состав модели СМО в обязательном порядке входят модели генераторов потока заявок и потока обслуженных заявок. В основу моделей генераторов положены источники последовательностей случайных чисел равномерно распределенных на интервале $(0 \dots 1)$. На основе этой последовательности можно имитировать потоки с различными законами распределения.

Пусть при моделировании некоторой системы необходимо сформировать на ЭВМ простейший поток заявок. Тогда длина интервала между $(i-1)$ -м и i -м событиями $y_i = -(1/\lambda) \ln(x_i)$.

Если при моделировании некоторой системы требуется сформировать на ЭВМ поток событий, равномерно распределенных на интервале (a, b) , то функция плотности интервалов между событиями $f(y) = 1/(b-a)$, $a \leq y \leq b$. Распределение первого интервала между началом отсчета и первым событием

$$f_1(y_1) = \lambda \left(1 - \int_0^{y_1} f(y) dy\right) = \lambda \left[1 - \int_0^{y_1} dy/(b-a)\right].$$

Интенсивность потока

$$\lambda = 1/M[y] = 1/\int_a^b y f(y) dy = 2/(a+b).$$

Тогда $f_1(y_1) = 2[1 - y_1/(b-a)]/(a+b)$.

где x_i — случайная величина, равномерно распределенная на интервале $(0, 1)$.

Для формирования на ЭВМ потока Эрланга, в котором между последовательными событиями закон распределения интервалов

$$f_k(t) = \frac{\lambda (\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, \quad t > 0.$$

плотность распределения длины первого интервала

$$f_1(y_1) = a'(t_0, y_1) e^{-a(t_0, y_1)},$$

где a — математическое ожидание числа событий на интервале $(t_0 + \Delta t)$, $a(t_0, y_1) = -\ln(x_1)$.

Примерный состав модели генератора представлен на рис.2

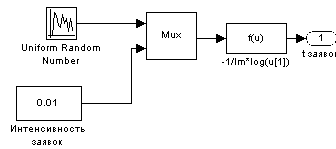


Рис.2 . Генератор

Функция, реализованная в генераторе, соответствует заданному закону распределения случайных величин. Генератор обслуженных заявок имеет аналогичную структуру.

Основной задачей накопителя является сравнение времени обслуживания заявки и времени поступления новой заявки. Если предыдущая заявка не была обслужена, формируется признак наличия задержанной заявки. На рис.3 показан примерный вид накопителя.

Вариант объединения накопителя и двух генераторов показан на рис.4.

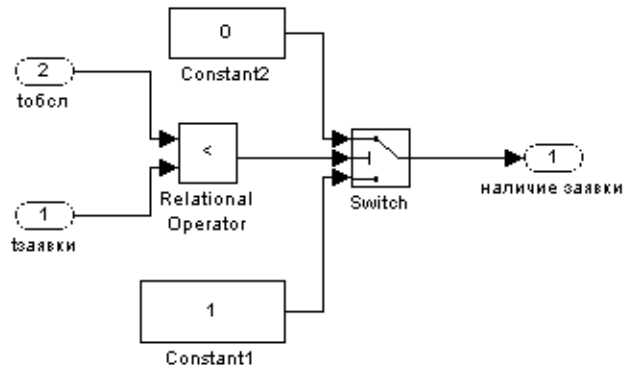


Рис. 3. Накопитель

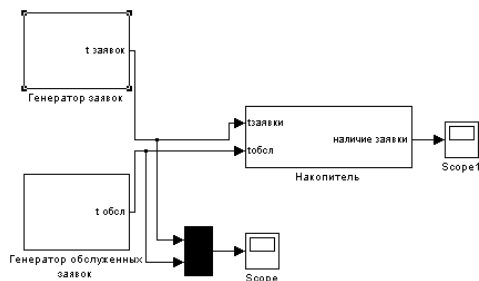


Рис. Пример соединения накопителя и генераторов

Лабораторная работа №4.

Исследование модели системы массового обслуживания

Цель занятия: Освоить методику моделирования элементов систем массового обслуживания (СМО)

В зависимости от характера источника заявок различают разомкнутые и замкнутые системы массового обслуживания СМО. На практическом занятии будут рассмотрены только разомкнутые СМО (рис.1). В зависимости от числа мест в очереди различают СМО с отказами и без отказов. В СМО с отказами число мест в очереди конечно и вследствие вероятностного характера, как входящего потока, так и процессов обслуживания, существует ненулевая вероятность того, что поступившая на вход СМО заявка застанет все каналы занятыми обслуживанием и все места в очереди занятыми заявками, ожидающими обслуживания, т. е. она получит отказ. В СМО без отказов заявка либо сразу назначается на обслуживание, если в момент ее поступления свободен хотя бы один канал, либо безусловно принимается в очередь.

1. Построение сетевых моделей одноканальных систем массового обслуживания

Кратко рассмотрим методику составления сетевых моделей на примере одноканальной СМО (в совокупности с генератором заявок). Следует отметить, что процесс разработки сетевых моделей в общем случае является неформализованным. Проинтерпретируем СМО в терминах транзакций и ресурсов. Транзакции – это активные подвижные элементы системы, а ресурсы – неактивные. Транзакциями в СМО являются заявки, а ресурсом является канал обслуживания заявок. Функционирование СМО описывается как взаимодействие транзакций и ресурсов.

При переходе от системы транзакций и ресурсов к сетевой модели можно пользоваться следующими правилами. Каждый ресурс представляется позицией, причем маркировка этой позиции (простыми метками) определяет состояние ресурса. В одноканальной СМО имеется один ресурс, имеющий два состояния: "занят" и "свободен", причем начальное состояние ресурса – "свободен". Поставим этому ресурсу в соответствие позицию R сетевой модели. Маркировка $M(R) = 1$ будет свидетельствовать о незанятости ресурса, а $M(R) = 0$ – о его занятости. Маркировка $M(R) \geq 2$ является запрещенной. Переход ресурса из одного состояния в другое представляется изменением маркировки позиции R : добавление метки соответствует освобождению ресурса, а изъятие – его занятию.

В системе транзакций и ресурсов каждая транзакция представляется простой меткой или меткой с атрибутами в зависимости от того, несет она информацию или нет. Будем считать, что в нашей одноканальной СМО каждая транзакция несет информацию и, следовательно, будет представляться меткой с атрибутами.

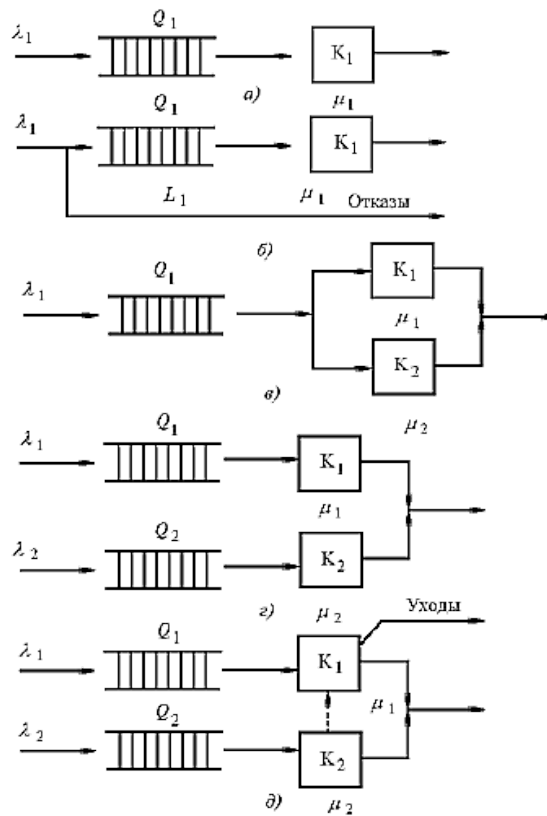


Рис. 1. Системы массового обслуживания:
а, б – одноканальные;
в, г, д – двухканальные

Транзакция в процессе ее обработки может находиться в нескольких состояниях, определяющих степень ее готовности. Для одноканальной СМО можно выделить следующие состояния транзакции:

- 0 – "готова для планирования";
- 1 – "планируется";
- 2 – "ожидание обслуживания";
- 3 – "заняла канал обслуживания";
- 4 – "обрабатывается";
- 5 – "обработана";
- 6 – "покинула канал обслуживания".

Состояния транзакции можно разделить на следующие классы:

- состояния временной активности (1, 4), связанные с процессами обработки транзакции, а также с планированием транзакции в генераторе транзакций. В этих состояниях транзакция находится во временной задержке перехода;
- состояния ожидания какого-либо ресурса (2). В этих состояниях транзакция находится в позициях, в которых нередко образуются очереди;
- состояния без ожидания (0, 3, 5, 6). В этих состояниях транзакция только "заявляет" о необходимости выполнения какого-либо события.

Каждому состоянию транзакции поставим в соответствие позицию или временную задержку перехода сетевой модели.

Переход транзакции из одного состояния в другое представляется как перемещение метки из одной позиции в другую, из позиции во временную задержку перехода, а также из временной задержки перехода в позицию.

В дальнейшем в моделируемой системе определяется множество событий, которые приводят к изменению состояния транзакций и ресурсов. В одноканальной СМО с генерацией заявок можно выделить следующие события:

1. "Приход новой заявки". При возникновении этого события транзакция переходит из состояния 1 в состояние 2. Создается копия транзакции, которая переводится в состояние 0.

2. "Планирование следующей заявки". Транзакция из состояния 0 переходит в состояние 1.

3. "Занятие канала обслуживания". Транзакция из состояния 2 переходит в состояние 3, а ресурс из состояния "свободен" переходит в состояние "занят".

4. "Начало обработки заявки". Транзакция из состояния 3 переводится в состояние 4.

5. "Конец обработки заявки". Транзакция из состояния 4 переводится в состояние 5.

6. "Освобождение канала обслуживания". Транзакция из состояния 5 переходит в состояние 6, а ресурс из состояния "занят" переходит в состояние "свободен".

7. "Уничтожение заявки". Транзакция покидает состояние 6 и уничтожается.

2. Моделирование очереди заявок

Очередью называется элемент исследуемого объекта, с помощью которого моделируются процессы ожидания начала обработки требований обслуживающими устройствами. Возникновение этих процессов происходит ввиду того, что в момент поступления требования устройство может находиться в состояниях "занято" либо "выключено". Кроме того, ожидание начала обработки может возникнуть из-за того, что приоритет поступившего требования меньше, чем приоритет обрабатываемого. По любой из вышеперечисленных причин, поступающие требования устанавливаются в очередь к устройству.

На входе устройства может формироваться произвольное количество очередей. При этом количество очередей не может превышать общего числа внесистемных и внутрисистемных источников, заданных в предложении *входящий поток*. В зависимости от количества этих источников и правил функционирования элементов, имитирующих процессы ожидания, для описания очередей используется одна из следующих конструкций:

очередь: общая / отдельная / смешанная;

С помощью первой конструкции формируется одна очередь, в которую могут устанавливаться все требования, поступающие на вход устройства. Остальные две конструкции позволяют формировать множество очередей перед устройством. Независимо от количества очередей, установленных на входе устройства, дисциплины их формирования определяются предложением *механизм обслуживания*.

Общая очередь

Общей очередью называется такая очередь, в которую могут устанавливаться требования, поступающие от всех внесистемных и внутрисистемных источников, указанных в предложении *входящий поток*. Задание общей очереди осуществляется с помощью конструкции *общая*. Формирование очереди производится независимо от способа передачи требований на вход устройства. Конкретное место, занимаемое в очереди каждым требованием, определяется в зависимости от его приоритета и механизма обслуживания. При этом основополагающим правилом является механизм обслуживания, устанавливающий дисциплину формирования очереди.

Пример модели обслуживающего устройства ($n,1$) с общей очередью представлен на рис. 1.

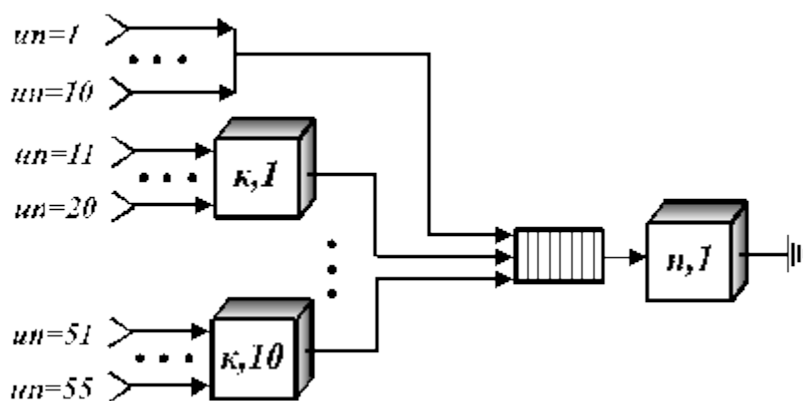


Рис. 1. Схема модели устройства с общей очередью

На вход устройства поступают требования от 10 внесистемных источников с индексами от 1 до 10 включительно и от 45 внутрисистемных источников. Интенсивность первых пяти источников требований равна 0.1, а следующих пяти источников - 0.2. Как показано на рисунке, на вход устройства $(n, 1)$ с устройства $(κ, 1)$ поступают требования от внутрисистемных источников с индексами от 11 до 20 включительно, а с устройства $(κ, 10)$ с индексами от 51 до 55 включительно. На устройствах $(κ, 2) - (κ, 9)$ обрабатываются требования, генерируемые внесистемными источниками с индексами от 21 до 50 включительно. Описание обслуживающего устройства с общей очередью можно представить следующим образом:

$у(n, 1); вп: (un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11-20 \text{ у}(κ, 1)), \dots (un=51-55 \text{ у}(κ, 10));$
од: общая; мо: впп;

С применением средств интегрированного описания, позволяющих использовать фиктивные связи, модель устройства $(n, 1)$ можно представить в виде:

$у(n, 1); вп: (un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11-55 \text{ у}(κ, 1)2-10); од: общая;$
мо: впп;

Во втором случае при наличии одного внутрисистемного входа во входящем потоке конструкция *од:общая*; также позволяет сформировать общую очередь на входе $(n, 1)$.

Отдельные очереди

Отдельными называются такие очереди, которые формируются автоматически для каждого внесистемного и внутрисистемного источника, заданного во входящем потоке.

При наличии нескольких очередей на входе устройства возникает вопрос об их приоритетах, который может решаться двумя способами: по правилу умолчания либо путем явного описания очередей в порядке убывания их приоритетов. Первый способ основывается на механизме обслуживания и порядке следования описаний внесистемных и внутрисистемных источников в предложении *входящий поток*. При этом используется конструкция *очередь: отдельная*. Установка приоритетов очередям осуществляется следующим образом. Вначале формируются очереди для всех внесистемных источников. Приоритеты этих очередей определяются порядком следования описаний внесистемных источников во входящем потоке. Затем строятся очереди для внутрисистемных источников. Приоритеты этих очередей всегда ниже приоритетов очередей, формируемых внесистемными источниками. Установка приоритетов таким очередям также осуществляется в порядке очередности появления соответствующих описаний внутрисистемных источников во входящем потоке. В процессе моделирования приоритеты очередей, установленные по правилу умолчания, могут изменяться. Подробно процедуры модификации этих приоритетов освещены при описании механизма обслуживания. Задание приоритетов очередям с использованием правила умолчания рассмотрим на примере модели, представленной на рис. 2.

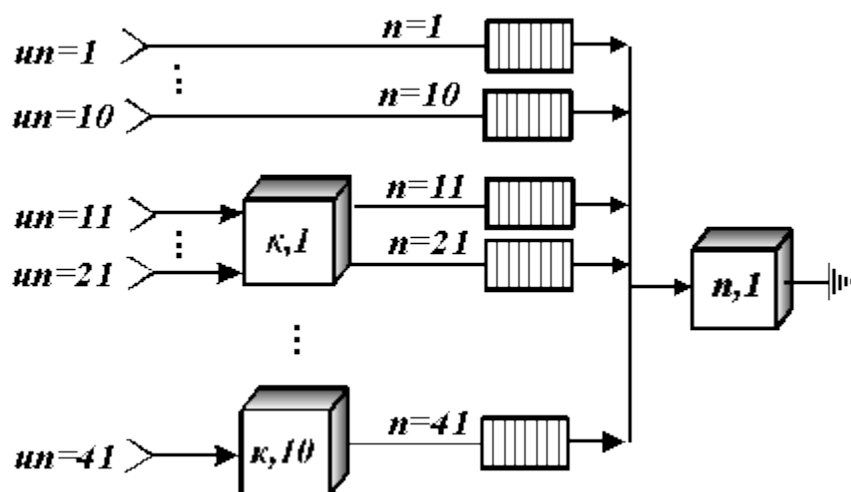


Рис. 2. Схема модели устройства с отдельными очередями

Как показано на схеме модели, на входе устройства (n, I) формируются отдельные очереди, образуемые десятью внесистемными потоками с индексами от 1 до 10 включительно и тридцатью внутрисистемными потоками с индексами от 11 до 41 включительно, поступающими с устройств (κ, I) - $(\kappa, 10)$. Описание этого фрагмента модели, иллюстрирующего формирование отдельных очередей, можно представить следующим образом:

$y(n, I); vn:(un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=11 \text{ у}(\kappa, 1)), (un=12 \text{ у}(\kappa, 1)), \dots (un=21 \text{ у}(\kappa, 1)), \dots (un=41 \text{ у}(\kappa, 10));$
од: отдельная;

Согласно приведенному описанию на входе устройства (n, I) формируются отдельные очереди. Каждый внутрисистемный источник описывается независимо, что позволяет сформировать сорок одну очередь.

Наиболее простой способ модификации умалчиваемых значений приоритетов отдельных очередей состоит в изменении последовательности описания источников требований во входящем потоке. Предположим, что в рассматриваемом примере очередь на входе (n, I) , формируемая 41 источником требований $(un=41 \text{ у}(\kappa, 1))$, должна быть наиболее приоритетной среди очередей, образующихся из требований, поступающих от внутрисистемных источников. Для реализации такой модели описание входящего потока должно быть выполнено следующим образом:

$y(n, I); vn:(un=1-5 \text{ э}(0.1)), (un=6-10 \text{ э}(0.2)), (un=41 \text{ у}(\kappa, 10)),$
 $(un=11 \text{ у}(\kappa, 1)), (un=12 \text{ у}(\kappa, 1)), \dots, (un=40 \text{ у}(\kappa, 9));$
од: отдельная;

Приоритеты очередей, формируемых внесистемными источниками, также определяются согласно очередности появления соответствующих описаний. Поэтому для того, чтобы очередь, формируемая десятым внесистемным источником была наиболее приоритетной среди остальных очередей, входящий поток необходимо описать следующим образом:

$y(n, I); vn:(un=10 \text{ э}(0.2)), (un=1-5 \text{ э}(0.1)), (un=6-9 \text{ э}(0.2)), (un=41 \text{ у}(\kappa, 10)), \dots;$

При использовании правила умолчания для задания приоритетов очередям применение средств интегрированного описания внесистемных и внутрисистемных источников имеет одну особенность. Она состоит в том, что независимо от степени интеграции описания внесистемного входа для каждого из перечисленных в нем источников строятся отдельные очереди, а для внутрисистемного входа всегда создается одна очередь.

В тех случаях, когда невозможно или нецелесообразно использовать неявный способ задания приоритетов очередям, можно применить второй способ установки приоритетов. Он предполагает задание приоритетов очередям путем последовательного описания соответствующих входов в предложении *очередь*:

отдельная

((<вход 1>) [<атрибуты очереди>])
[,(<вход 2>) [<атрибуты очереди>]]
[...,((<вход N>)[<атрибуты очереди>])]]

Представленная синтаксическая конструкция позволяет первой очереди, задаваемой с помощью конструкции <вход 1>, установить наивысший приоритет, а последней очереди, <вход N> - наименьший. Например, конструкция **од: отдельная** ((*un*=4)), ((*un*=11-15 *y*(*κ*,1))), ((*un*=8-10)); устанавливает убывающие приоритеты очередям, формируемым различными типами источников в следующем порядке:

- 1 очередь - ((*un*=4));
- 2 очередь - ((*un*=11-15 *y*(*κ*,1)));
- 3 очередь - ((*un*=8));
- 4 очередь - ((*un*=9));
- 5 очередь - ((*un*=10));

Из приведенного примера видно, что для каждого внесистемного источника всегда формируется отдельная очередь, хотя в тексте модели задание третьей, четвертой и пятой очередями осуществляется с помощью интегрированного входа ((*un*=8-10)).

При построении отдельных очередей не обязательно описывать все источники требований, заданные во входящем потоке. В этом случае алгоритм построения очередей и установки им необходимых приоритетов будет основываться на использовании двух указанных выше способов. Наиболее приоритетными будут очереди, формируемые на основе их явного описания. Для неописанных источников очереди создаются по правилу умолчания. Предположим, что описание входящего потока и очереди для устройства (*n*,1) представлено следующим образом:

вп: ((*un*=11-21 *y*(*κ*,1)), (*un*=22-24 *y*(*κ*,2)), (*un*=25-26 *y*(*κ*,3)),
((*un*=27-28 *y*(*κ*,4)), (*un*=29-30 *y*(*κ*,5)), (*un*=31-32 *y*(*κ*,6)),
((*un*=33-34 *y*(*κ*,7)), (*un*=35-36 *y*(*κ*,8)), (*un*=37-40 *y*(*κ*,9)),
((*un*=41 *y*(*κ*,10)),(*un*=1-10 э(0.01));
од: **отдельная** ((*un*=25-26 *y*(*κ*,3))), ((*un*=1)), ((*un*=11-21 *y*(*κ*,1)));

Согласно этому описанию наивысшим приоритетом будет обладать очередь, формируемая из требований, поступающих с устройства (*κ*,3). Второй по порядку приоритет будет иметь очередь, образуемая из требований потока с индексом 1. Следующим по порядку приоритетом будет обладать очередь, формируемая из требований потоков с индексами от 11 до 21 включительно. Приоритеты остальным очередям задаются по правилу умолчания.

Смешанные очереди

Конструкция **смешанная** предназначена для формирования очередей, в которые могут устанавливаться требования, поступающие от различных типов источников. С помощью этой конструкции пользователю предоставляется возможность задавать различные правила объединения потоков для формирования очереди. Вслед за конструкцией **смешанная** необходимо указать, из каких источников должна формироваться первая очередь, затем вторая и т.д. Задание правила формирования очереди производится путем перечисления входов, образующих эту очередь. Далее может следовать описание атрибутов очереди, определяющих некоторые особенности ее функционирования. Синтаксис смешанной очереди представляется следующим образом:

смешанная

((<вход>)[,<вход>][...,<вход>] [<атрибуты очереди>]]]
[,(<вход>)[,<вход>][...,<вход>] [<атрибуты очереди>]]]
[...,((<вход>)[,<вход>][...,<вход>])] [<атрибуты очереди>]]]

Каждая из очередей, формируемых с использованием конструкции **смешанная**, может образовываться из внесистемных и внутрисистемных источников. Ниже приведен пример модели, в которой на входе устройства (*n*,1) имеются две очереди (рис. 3).

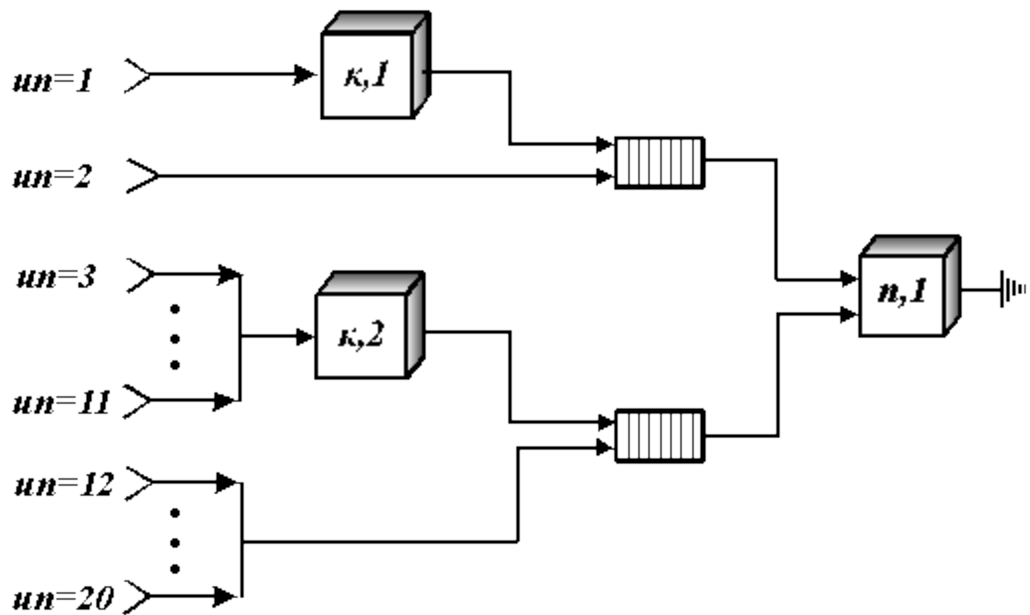


Рис. 3. Схема модели устройства со смешанными очередями

Первая из них формируется внутрисистемным потоком с индексом 1, поступающим с устройства ($\kappa,1$), и внесистемным потоком с индексом 2. Вторая очередь также образуется из различных типов источников. При этом внутрисистемные потоки с индексами 3 - 11 поступают с устройства ($\kappa,2$). Кроме того, в очередь могут устанавливаться требования, поступающие от девяти внесистемных источников с номерами 12 - 20. Описание фрагмента модели, иллюстрирующего задание смешанных типов очередей, представляется следующим образом:

$y(n,1)$; $вп$: ($un=1 y(\kappa,1)$), ($un=2 \varepsilon(0.1)$), ($un=3-11 y(\kappa,2)$), ($un=12-20 \varepsilon(0.2)$);
од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-20$)));

Если в описании смешанной очереди перечислены не все входы, то при построении очередей по умолчанию для каждого из них создается отдельная очередь. Предположим, что в рассмотренной выше модели для девятнадцатого и двадцатого потоков необходимо построить отдельные очереди. Для этого достаточно при описании правила формирования второй очереди опустить в перечислении девятнадцатый и двадцатый потоки:

$y(n,1)$; $вп$: ($un=1 y(\kappa,1)$), ($un=2 \varepsilon(0.1)$), ($un=3-11 y(\kappa,2)$), ($un=12-20 \varepsilon(0.2)$);
од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-18$)));

Этой же цели можно достигнуть путем прямого задания правила формирования третьей и четвертой очередей для 19-го и 20-го потоков:

$y(n,1)$; $вп$: ($un=1 y(\kappa,1)$), ($un=2 \varepsilon(0.1)$), ($un=3-11 y(\kappa,2)$), ($un=12-20 \varepsilon(0.2)$);
од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-18$)), (($un=19$)), (($un=20$)));

Если необходимо, чтобы 19 и 20-ый потоки образовывали одну очередь, то описание очереди должно быть представлено следующим образом:

од: смешанная (($un=1 y(\kappa,1)$), ($un=2$)), (($un=3-11 y(\kappa,2)$), ($un=12-18$)), (($un=19$), ($un=20$)));

Как и в случае с отдельным типом очереди, при наличии более одной очереди на входе устройства, возникает вопрос об их приоритетах. Приоритет задается согласно очередности описания правила формирования очереди. Первая очередь, встретившаяся в описании, имеет наивысший приоритет. Приоритет остальных очередей последовательно убывает. Если при описании правил формирования смешанной очереди указаны не все входы, перечисленные во входящем потоке, то для них по умолчанию также создаются отдельные очереди. При этом сначала строятся очереди для внесистемных, а затем - для внутрисистемных потоков.

СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

	Авторы,	Заглавие	Издательство, год	Адрес	
Л1.1	Блинков, Ю. В.	Основы теории информационных процессов и систем: учебное пособие	Пенза: Пензенский государственный университет архитектуры и строительства, ЭБС АСВ, 2011	http://www.iprbookshop.ru/23103.html	
Л1.2	Белов, П. С.	Математическое моделирование технологических процессов: учебное пособие (конспект лекций)	Егорьевск: Егорьевский технологический институт (филиал) Московского государственного технологического университета «СТАНКИН», 2016	http://www.iprbookshop.ru/43395.html	
6.1.2. Дополнительная литература					
	Авторы,	Заглавие	Издательство, год	Адрес	
Л2.1	Казиев В. М.	Введение в анализ, синтез и моделирование систем	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/52188.html	
Л2.2	Плохотников, К. Э.	Методы разработки математических моделей и вычислительный	Москва: СОЛОН-ПРЕСС, 2017	http://www.iprbookshop.ru/64926.html	
6.1.3. Методические разработки					
	Авторы,	Заглавие	Издательство, год	Адрес	
Л3.1	Татарникова, Т. М.	Моделирование систем: методические указания к выполнению лабораторных работ	Санкт-Петербург: Российский государственный гидрометеорологический университет, 2008	http://www.iprbookshop.ru/12503.html	
Л3.2	Шевцова, Ю. В.	Математические модели и методы исследования операций: сборник задач	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2009	http://www.iprbookshop.ru/54766.html	
Л3.3	Сёмина, В. В.	Моделирование систем: методические указания для проведения лабораторных работ по дисциплине «моделирование	Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2016	http://www.iprbookshop.ru/64869.html	
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"					
Э1	Душин В.К.	Теоретические основы информационных процессов и систем [Электронный ресурс]: учебник/ В.К.— Электрон. текстовые данные.— М.: Дашков и К, 2014.— 348 с.— Режим доступа: http://www.iprbookshop.ru/24764 .— ЭБС «IPRbooks», по паролю			
Э2	Шатрова Г.В.	Методы исследования и моделирования информационных процессов и технологий [Электронный ресурс]: учебное пособие/ Шатрова Г.В., Топчиев И.Н.— Электрон. текстовые данные.— Ставрополь: СКАУ, 2016.— 180 с			

ЭЗ	Лубенец Ю.В. Экономико-математические методы и модели [Электронный ресурс]: учебное пособие/ Электрон. текстовые данные.— Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013.	Лубенец Ю.В.
----	--	--------------