

Федеральное агентство по образованию  
Нижегородский государственный университет им. Н.И. Лобачевского

Национальный проект «Образование»  
Инновационная образовательная программа ННГУ. Образовательно-научный центр  
«Информационно-телекоммуникационные системы: физические основы и  
математическое обеспечение»

Ю.А. Кузнецов, В.И. Перова

**Применение пакетов имитационного  
моделирования для анализа математических моделей  
экономических систем**

*Учебно-методические материалы по программе повышения квалификации  
«Применение программных средств в научных исследованиях и  
в преподавании математики и механики»*

Нижний Новгород

2007

Учебно-методические материалы подготовлены в рамках  
инновационной образовательной программы ННГУ:  
*Образовательно-научный центр*  
*«Информационно-телекоммуникационные*  
*системы: физические основы и математическое обеспечение»*

Кузнецов Ю.А., Перова В.И. Применение пакетов имитационного моделирования для анализа математических моделей экономических систем: Учебно-методический материал по программе повышения квалификации «Применение программных средств в научных исследованиях и в преподавании математики и механики». Нижний Новгород, 2007. 98 с.

Учебное пособие содержит описание пакетов имитационного моделирования, практические примеры с применением пакета Ithink и предназначено для студентов, обучающихся по специальности «Математические методы в экономике», по специальностям в области экономики и прикладной математики, а также адресовано аспирантам, преподавателям и научным сотрудникам.

©Ю.А. Кузнецов, В.И. Перова, 2007

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ</b>	5
<b>ГЛАВА 1. ПАКЕТЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В АНАЛИЗЕ ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ</b>	6
1.1. Сравнение пакетов имитационного моделирования	7
1.1.1. Пакет Process Charter	10
1.1.2. Пакет Powersim	11
1.1.3. Пакет Ithink	12
1.1.4. Пакет Extend + BPR	14
1.1.5. Пакет Pilgrim	16
1.1.6. Пакет Vensim	17
<b>ГЛАВА 2. РАБОТА С ПРОГРАММНЫМ ПАКЕТОМ ITHINK</b>	19
2.1. Основные операции	21
2.1.1. Уровни представления моделей	21
2.2.1.1. Уровень интерфейса	21
2.2.1.2. Уровень модели	22
2.2.1.3. Уровень программного кода	23
2.2. Меню	24
2.2.1. Меню File	24
2.2.2. Меню Edit	25
2.2.3. Меню Interface, Model and Equation	26
2.2.4. Меню Run	31
2.2.5. Меню Help	35
2.3. Основные строительные блоки	35
2.3.1. Строительные блоки на уровне интерфейса	35
2.3.1.1. Фрейм	35
2.3.1.2. Поточковая связь	37
2.3.1.3. Коннекторная связь	38
2.3.2. Строительные блоки на уровне модели	39
2.3.2.1. Фонды	39
2.3.2.2. Резервуар	39
2.3.2.3. Конвейер	41
2.3.2.4. Очередь	47
2.3.2.5. Печь	49
2.3.2.6. Поток	51
2.3.2.7. Конвертер	55
2.3.2.8. Коннектор	58
2.3.2.9. Ромбоидальное определение процесса	60
2.4. Инструменты	63
2.4.1. Инструмент «Перо»	63
2.4.2. Инструмент «Кисть»	63
2.4.3. Инструмент «Динамит»	63
2.4.4. Инструмент «Фантом»	63
2.5. Объекты	64
2.5.1. Фрейм сектора	65
2.5.2. Текст	66

2.5.3. Рисунок	66
2.5.4. Числовой дисплей	67
2.5.5. График	68
2.5.6. Таблица	71
<b>ГЛАВА 3. ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ АНАЛИЗА ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ</b>	74
3.1. Влияние изменения процентной ставки на сальдо банковского счета	74
3.2. Модель распределения доходов фирмы между доходами от продажи и сервисным обслуживанием	79
3.3. Модель кредитования трех предприятий	88
<b>ЛИТЕРАТУРА</b>	98

## **ВВЕДЕНИЕ**

Цель данного учебного пособия – освещение применения имитационного моделирования в социально - экономической деятельности и современных информационных технологий, предоставляющих различные возможности для анализа математических моделей социальных и экономических систем.

Имитационную модель необходимо создавать. Для этого нужно специальное программное обеспечение – система моделирования. Специфика такой системы определяется технологией работы, набором языковых средств, сервисных программ и приемов моделирования. Имитационное моделирование контролируемого процесса или управляемого объекта – это высокоуровневая информационная технология, которая обеспечивает работы по созданию или модификации имитационной модели, а также эксплуатацию имитационной модели и интерпретацию результатов.

Сложность реальных социально – экономических процессов приводит к тому, что для имитационного моделирования этих процессов применяются достаточно технологичные инструментальные средства имитационного моделирования, обладающие собственными языковыми средствами. Обзор таких средств приведен в главе 1.

В главе 2 представлена подробная инструкция для пользователя по работе с пакетом имитационного моделирования ITHINK.

Глава 3 посвящена практическому применению технологии имитационного моделирования с использованием пакета ITHINK для решения социально - экономических задач.

## ГЛАВА 1. ПАКЕТЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В АНАЛИЗЕ ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ

Развитие средств имитационного моделирования связано с необходимостью исследования очень сложных систем. Имитация – это своеобразная попытка дублировать особенности, внешний вид и характеристики реальной системы. Применение имитационного моделирования оправдано, если вопросы, ответ на которые должна дать модель, относятся не к выяснению фундаментальных законов и причин, определяющих динамику реальной системы, а к практическому анализу поведения системы [1-4].

*«Имитационное моделирование – это распространенная разновидность аналогового моделирования, реализуемого с помощью набора математических инструментальных средств, специальных имитирующих компьютерных программ и технологий программирования, позволяющих посредством процессов-аналогов провести целенаправленное исследование структуры и функций реального сложного процесса в памяти компьютера в режиме «имитации» и выполнить оптимизацию некоторых его параметров» [5].*

При достаточно глубоком знании поведения реальной системы и правильном представлении исходной информации имитационные модели характеризуются большей близостью к реальной системе, чем аналитические и численные модели. С помощью имитационного моделирования и соответствующих современных программных средств можно создавать даже те модели, которые не возможно сформулировать традиционными методами. Кроме того, имитационное моделирование позволяет создавать модели тех систем, с которыми нельзя провести эксперимент, что позволяет упростить и усовершенствовать управление ими.

Однако имитационное моделирование обладает и рядом недостатков. Основным из них и практически непреодолимым является то, что имитация – это не наука и не искусство, а некая их комбинация. Кроме того, имитационное моделирование является итеративной, экспериментальной техникой решения проблем; требует значительных затрат кадров и компьютерного времени; чаще всего находит решение, только близкое к оптимальному. Ученому трудно оценить адекватность модели и найти источник ошибок; а сбор, анализ и интерпретация результатов требует хорошего знания теории вероятности и статистики. Однако очень часто этот метод является единственным способом создания модели сложной системы.

Впервые методы имитационного моделирования для анализа экономических процессов применил Т. Нейлор. На протяжении 1970-1980-х годов попытки использовать этот вид моделирования в реальном управлении экономическими процессами носили эпизодический характер из-за сложности формализации экономических процессов и недостаточной развитости программных средств. Однако именно в середине 1970-х гг. появились первые инструментальные средства имитационного моделирования, например, система GPSS, которая позволяла создавать модели контролируемых процессов и объектов в основном технического или технологического назначения [3].

В 80-х годах прошлого века системы имитационного моделирования стали развиваться более активно. В то время в различных странах мира применялось более 20 различных систем (GASP-IV, SIMYLA-67, SLAM-II, GPSS-V и т.д.). В 90-х годах появилось множество пакетов для имитационного моделирования, которые становились все более развитыми и удобными даже для неподготовленного пользователя. Такой расцвет можно объяснить не только увеличением интереса к имитационному моделированию, но и большим прогрессом в развитии компьютерной техники, быстродействие и возможности которой возрастают с каждым днем. Среди таких средств можно выделить Powersim, Ithink, Process Chapter, Simula и т.д.

### **1.1. Сравнение пакетов имитационного моделирования**

Чтобы лучше понять, как работают пакеты имитационного моделирования, рассмотрим в качестве примера центр обслуживания, подобный тому, который функционирует в банке, справочной службе или на предприятии по ремонту автомобилей. Целью менеджера является снижение затрат путем уменьшения до минимума количества обслуживающего персонала - кассиров банка, операторов телефона или автомобильных механиков, причем время, потраченное клиентами на ожидание, должно быть как можно меньше. Чтобы добиться этого, менеджеру необходимо получить от модели следующую информацию: количество обслуживающего персонала, число клиентов в очереди на обслуживание и время ожидания.

Для моделирования процесса обслуживания клиентов, прежде всего, требуется, чтобы специалист, создающий модель, с помощью определенной среды разработки, входящей в данное средство моделирования, построил диаграмму системы, через которую будет проходить поток входных данных. В центре обслуживания входными данными являются число клиентов, чьи нужды следует удовлетворить, частота их появления в центре, количество обслуживающего персонала и время, потраченное служащим на одного

клиента. Частота появления клиентов вводится в один блок модели, а количество обслуживающего персонала и время обслуживания клиента - в другой. В ходе исполнения модели клиенты переходят из блока в блок со скоростью, соответствующей времени обслуживания.

Завершив построение модели, разработчик может определить зависимость между числом клиентов, обратившихся за обслуживанием, и временем ожидания в очереди, а также выяснить, сколько клиентов оказались необслуженными. Затем полученную информацию можно сопоставить с требованиями, предъявляемыми к данному центру обслуживания, и допустимыми ограничениями, а затем в интерактивном режиме ввести в модель некоторые изменения, например, увеличить количество обслуживающего персонала и снова запустить модель на исполнение и так далее до получения удовлетворительного результата.

Существует два подхода к имитационному моделированию - *статический и динамический*. Статические модели представляют собой системы уравнений, которые решаются один раз. Динамические модели включают в себя еще одну переменную - время. Математические расчеты параметров бизнес - процессов выполняются на различных временных интервалах, позволяя тем самым разработчику модели исследовать развитие системы во времени. Поскольку большинство бизнес - процессов зависит от времени, рассмотрим только те пакеты, которые обеспечивают проведение *динамического моделирования*.

В свою очередь, динамические модели бывают двух типов: *непрерывные и дискретные*. В непрерывных моделях время изменяется линейно, а процесс - в непосредственной зависимости от времени. Непрерывная модель позволяет моделировать загрязнение окружающей среды или поток жидкости в трубопроводе. В дискретных моделях переменными являются события и временные интервалы. Такие модели, например, дают возможность оценивать производительность компьютера или управлять складскими запасами.

Центр обслуживания, описанный выше, представляет собой *динамическую дискретную модель*, в которой через определенные интервалы времени в процесс поступают клиенты, становятся в очередь, получают обслуживание и удаляются. Событием, которое управляет моделью, служит появление клиентов. Для оценки возможностей пакетов имитационного моделирования необходимо учитывать простоту разработки и исполнения модели, количество заранее сформированных компонентов, набор форматов для вывода результатов и документацию. Кроме того, нужно оценить



дополнительные средства, которые позволяют технически грамотным программистам строить собственные модели и снабжать их специальными элементами управления, чтобы менее подготовленные пользователи могли модифицировать и исполнять эти модели.

Сравнительная характеристика [3, 6] наиболее известных пакетов имитационного моделирования представлена в таблице 1.1.

Таблица 1.1

Сравнение пакетов имитационного моделирования

Система моделирования	Производитель	Приложения	Моделирующая среда и поддержка			
			Графическая конструкция ИМ	Авторское моделирование, программирование моделей	Анимация (в реал. времени)	Поддержка анализа результатов
EXTEND	Imagine That, Inc.	Стратегическое планирование, бизнес-моделирование	Компоновочные блоки, непрерывные и дискретные модели	+ язык progr. Modl	+	Анализ чувствительности
ITHINK	High Performance System, Inc	Управление финансовыми потоками, реинжиниринг предприятий, банков, инвестиционных компаний и др.	CASE-средства, потоковые диаграммы	+	+	Анализ чувствительности
VENSIM	Ventana Systems	Модели системной динамики	Потоковые диаграммы	—	+	+
POWERSIM	Powersim Co.	Непрерывное моделирование	Потоковые диаграммы	—	+	—
PILGRIM	МЭСИ, Россия	Модели динамических систем	Компоновочные блоки, CASE-средства	Язык progr. C++	+	+
PROCESS CHARTER	Scitor	Дискретное моделирование	Блок-схемы	—	+	—

Рассмотрим подробнее следующие пакеты имитационного моделирования [3, 6]:

- **Process Charter 1.0.2** компании Scitor (Менло-Парк, шт. Калифорния);
- **Powersim 2.01** фирмы Modell Data AS (Берген, Норвегия);
- **Ithink 3.0.61** производства High Performance Systems (Ганновер, шт. Нью-Хэмпшир);
- **Extend+BPR 3.1** компании Imagine That! (Сан-Хосе, шт. Калифорния);
- **Pilgrim** производства МЭСИ, Россия;

- **Vensim** фирмы Ventana Systems.

Эти продукты более всего различаются стилем моделирования, т. е. средой, посредством которой создаются модели. В пакете **Process Charter** модель строится с помощью блок-схемы. **Powersim** и **Ithink** используют систему обозначений **Systems Dynamics**, предложенную в 1961 г. Джейм Форрестером (Jay Forrester) из Массачусетского технологического института. И, наконец, **Extend** применяет компоновочные блоки. Все продукты, кроме **Process Charter**, позволяют проводить анализ чувствительности, т. е. многократно исполнять модель с различными входными параметрами, чтобы сравнить результаты нескольких прогонов.

### 1.1.1. Пакет **Process Charter**

Компания Scitor назвала пакет **Process Charter** интеллектуальным средством создания *блок-схем*. Поскольку большинство пользователей знакомы с принципами построения блок-схем, разработка базовой модели не вызывает трудностей, так как напоминает создание блок-схемы любого процесса. В то же время другие рассматриваемые нами пакеты содержат более развитые средства имитационного моделирования. Поэтому **Process Charter**, хотя и позволяет успешно проводить дискретное моделирование, особенно в тех случаях, когда требуется анализ стоимости, малоэффективен при построении непрерывных моделей и выходных диаграмм.

Компания Scitor включила в свой продукт множество типов блоков, стилей строк и видов текста для форматирования диаграмм моделей. Наличие различных средств форматирования, таких как заголовки, полутона, цвета, жирный шрифт и инструментальные линейки с большим количеством кнопок, делает **Process Charter** похожим на любой другой **Windows-пакет** построения блок-схем. Однако на этом сходство заканчивается. Каждый из блоков, получивших название *Activity* (действие), имеет ряд атрибутов, специально предназначенных для моделирования. Первый (или исходный) блок управляет потоком элементов, проходящим через систему. Такая организация указывает на то, что *пакет Process Charter в основном ориентирован на дискретное моделирование. Однако благодаря тому, что модели, созданные с помощью пакета Process Charter, изменяются во времени с постоянной скоростью, продукт можно настроить и на непрерывное моделирование.*

Построив блок-схему, описывающую физическую основу модели, разработчик, используя электронную таблицу *Resource*, определяет ресурсы, требуемые для модели. Например, при моделировании гипотетического центра обслуживания можно задать три

очереди и снабдить каждую из них обслуживающим оператором в качестве одного из ресурсов. При определении ресурсов необходимо указать их тип (работник или материал) и характеристику (расходуемый или многократно используемый), а также ввести атрибуты стоимости и количества. Наличие атрибута стоимости делает **Process Charter** подходящим средством для анализа затрат, а наличие атрибута количества позволяет вручную увеличивать число операторов и проводить исследование чувствительности, многократно запуская модель. *Однако, из-за того что пакет **Process Charter** лишен специальных элементов управления, он не предоставляет возможности строить авторские модели.* Процесс присвоения ресурсов действиям организован по принципу *облокнотам*, в котором представлены связи между действиями, и разработчик может даже указать атрибуты для каждой из них. Во время исполнения модели прохождение ресурсов по блокам действий представляется на экране посредством *цветной анимации*. Это позволяет разработчику в оперативном режиме проводить предварительный анализ процесса. Чтобы просмотреть выходную информацию, разработчик может обратиться либо к странице результатов в блокноте действий, либо к одной из пяти электронных таблиц. Наиболее важные выходные значения можно поместить в электронную таблицу *Key Values* для коллективного просмотра. Кроме того, **Process Charter** содержит 13 заранее сформированных гистограмм для просмотра информации о моделировании. К сожалению, в пакете не предусмотрено средство, позволяющее пользователям строить собственные гистограммы, применять различные типы диаграмм, такие как линейные, круговые и диаграммы разброса, а также включать в гистограммы другие данные.

Недостатки пакета **Process Charter**:

- небольшое количество возможностей;
- слабая поддержка моделирования непрерывных компонентов;
- ограниченный набор средств для анализа чувствительности и построения диаграмм.

### 1.1.2. Пакет Powersim

Пакет **Powersim** является прекрасным средством создания *непрерывных* моделей. Однако с точки зрения дискретного моделирования он недостаточно эффективен. **Powersim** подходит пользователям, которым требуется построение непрерывных моделей, и которые хотят изучить довольно сложную систему обозначений **Systems Dynamics**.

Эта система реализована в **Powersim** в виде следующих конструкций. Первая, называемая *уровнем*, состоит из таких компонентов, как деньги, складские запасы, вредные выбросы и т. д. Вторая конструкция – *поток*. Он объединяет элементы, которые

перемещаются между уровнями. Вспомогательные атрибуты, напоминающие ячейки формул в электронных таблицах, и константы позволяют модифицировать поток. В процессе построения модели разработчик размещает блоки и определяет переменные, которые составляют математическую основу каждого блока. В этом ему помогает диалоговое окно *Define Variable*, в котором содержится список всех допустимых переменных для каждого блока и поле для описания переменных, чтобы пользователи легче могли понять эту модель. **Пакет Powersim выделяется среди остальных пакетов способностью обрабатывать массивы и поддерживать коллективную работу, а также тем, что содержит библиотеку с большим числом функций.** Массивы, позволяющие разработчику использовать наборы переменных, особенно удобны при построении моделей со сходной структурой. Например, при моделировании работы завода, имеющего 10 линий для производства красок различных цветов, разработчик может просто смоделировать одну линию и присвоить ей массив с информацией о цветах краски. Это значительно упростит построение модели и облегчит ее понимание. Массивы удобны и для создания моделей, в конструкции которых уровни меняют свое состояние, а разработчик хочет проследить за этими изменениями. **Powersim** включает в себя более 150 функций, разделенных на 16 групп, в том числе финансовую, математическую, статистическую, графическую и историческую. Подобно другим пакетам, **Powersim** использует при исполнении моделей *средства анимации*. Ключевые параметры, диаграммы и таблицы можно выводить непосредственно на экран моделирования, упрощая тем самым просмотр результатов. Функция *Multiuser Game* предоставляет возможность нескольким пользователям одновременно запускать модель, чтобы совместно над ней работать. Это особенно полезно для рабочих коллективов, проводящих тестирование.

**Powersim** содержит многие стандартные средства **Windows- приложений**, такие как *меню и инструментальные линейки*, и поддерживает технологии *Dynamic Data Exchange (DDE) и Object Linking and Embedding (OLE)*. Например, с помощью *OLE* разработчик может встроить **Powersim**-модель в документ, созданный текстовым процессором, так, чтобы изменения в модели автоматически отражались в документе. Пакет предоставляет множество примеров моделей, что позволяет новичку быстро научиться работать с ним и использовать все его возможности.

Недостатки пакета **Powersim**:

- ограниченная поддержка дискретного моделирования.

### 1.1.3. Пакет **Ithink**

Пакет **Ithink** – один из наиболее мощных продуктов, рассматриваемых нами. *С точки зрения непрерывного моделирования он отстает от Powersim, однако лучше поддерживает дискретное моделирование.* Кроме того, пакет **Ithink** снабжен превосходными обучающей программой и документацией, а также большим количеством блоков для составления модели. Пакет выпускается в двух версиях – **Basic** и **Authoring**. Версия **Authoring**, которая сравнивалась с другими пакетами, позволяет разработчику включать в модель *линейки с движками* и другие средства управления моделью, а также *вводить диаграммы и прочие изображения прямо в модель*, чтобы пользователи могли контролировать процесс моделирования и сразу видеть его результаты.

Подобно **Powersim**, пакет **Ithink** использует систему обозначений **Systems Dynamics**, которая в основном ориентирована на *непрерывное моделирование*. Для реализации этой системы служат конструкции четырех типов: *станции*, аналогичные уровням в пакете **Powersim**, *потоки, конвертеры*, напоминающие вспомогательные атрибуты из **Powersim**, и *соединители*, соответствующие связям. Чтобы создавать *дискретные* модели, **Ithink** использует три специальные станции:

- *очереди*, в которых элементы обрабатываются по принципу первым пришел – первым обслужен;
- *хранилища*, которые перед началом обслуживания накапливают заданное количество элементов и удобны при пакетной обработке;
- *транспортеры*, которые передают элементы между станциями.

Модели, построенные с помощью **Ithink**, состоят из уровней и иерархий. Пользователь строит описание модели на высоком уровне с помощью сред моделирования процессов, каждая из которых позволяет создать модель одной подсистемы, например такой, как расход ракетного топлива в космическом корабле. Завершив описание, разработчик переходит на следующую ступень детализации и вводит в каждую подмодель необходимые конструкции. Между подмоделями устанавливаются связи, указывающие на взаимодействие подсистем, к примеру, на зависимость между тем, насколько правильно компания составляет счета для оплаты, и количеством телефонных звонков в отдел обслуживания покупателей. Построив модель, снабженную необходимым числом иерархических уровней, разработчик переходит в режим моделирования, чтобы определить математические связи между станциями, потоками и другими конструкциями. Подобно **Powersim**, пакет **Ithink** предлагает разработчику список допустимых

переменных для определения математических связей. **Ithink** обеспечивает *проведение анализа чувствительности* модели путем ее многократного запуска с различными входными параметрами. Результаты каждого прогона выводятся в отдельной строке выходной диаграммы. Исходными данными являются основные виды распределений, применяемые для статистического анализа или диаграммы. При исполнении модели **Ithink** использует *средства анимации*, перемещающие расположенные на различных уровнях станции в соответствии с логикой модели.

Результаты моделирования выводятся в виде *временных диаграмм* или *диаграмм разброса*. Чтобы задать диаграмму или таблицу, разработчику требуется только выбрать используемые величины и указать необходимые параметры. Таблицы и диаграммы можно просматривать в специальных окнах пакета или в окне модели. *Хотя выбор форматов для вывода результата в Ithink не настолько широк, как в Extend, с этой точки зрения он превосходит как Powersim, так и Process Charter.* Компания High Performance Systems снабдила пакет **Ithink** *руководством по моделированию под названием «Системное мышление» и специальной информацией, предназначенной для пользователей, работающих в различных отраслях бизнеса.*

Недостатки пакета **Ithink**:

- поддерживает меньшее количество функций, чем Powersim.

#### 1.1.4. Пакет **Extend+BPR**

Пакет **Extend** как универсальное средство моделирования удобен при реорганизации различных бизнес – процессов. Для создания моделей в пакете применяется *блочная среда разработки*, пользоваться которой намного проще, чем системой обозначений **Systems Dynamics** для пакетов **Powersim** и **Ithink**, хотя несколько сложнее, чем блок-схемами продукта **Process Charter**. *Пакет Extend, обладающий средством построения непрерывных и дискретных моделей, широким диапазоном заранее сформированных блоков, поддержкой сторонних поставщиков и возможностью расширения является самым мощным продуктом из рассматриваемых нами.*

**Extend** впервые появился на рынке в 1987 г., поэтому его нельзя назвать новым продуктом. Вначале он был ориентирован на удобный пользовательский интерфейс компьютеров **Macintosh**, затем перенесен в среду **Windows** с помощью интерфейса прикладного программирования **Win32** и теперь даже выполняет инсталляцию **Win32** на системах, которые не имеют **Win32**. Пакет выпускается в четырех версиях: **Basic**, **Extend+BPR** (Business Process Reengineering), **Extend+Manufacturing** и **Extend+BPR+Manufacturing**. *Дополнительные средства BPR и Manufacturing*

включают ряд функций для вертикальных рынков. Кроме того, существует множество продуктов сторонних компаний, поддерживающих **Extend** и ориентированных на определенные области применения. К наиболее популярным из них относятся изделия, предназначенные для комитета **NASA**, управляющих систем, реорганизации бизнес-процессов (**BPR**), а также для массового производства.

В базовый пакет **Basic** входит более 90 заранее сформированных блоков, объединенных в библиотеки, из которых чаще других используются **Discrete-Event**, **Generic** и **Plotter**. Библиотека **Discrete-Event** включает в себя различные действия, очереди, иллюзы и таймеры. Библиотека **Generic** содержит генераторы случайных чисел и исходных данных, файлы для входной и выходной информации, а также блоки для математических, булевых и финансовых данных. Библиотека **Plotter** состоит из блоков для создания выходных диаграмм и таблиц. Остальные библиотеки имеют специальное назначение, например, осуществляют сбор статистической информации.

Пакеты **BPR** и **Manufacturing** снабжены дополнительными библиотеками. Кроме того, в **Extend** имеется встроенный язык **Modl**, позволяющий разработчику строить специализированные блоки. Выбором блока из библиотеки **Discrete-Event** автоматически строится дискретная модель; в противном случае – непрерывная модель. Блоки могут снабжаться входными и выходными соединителями. При построении связей не требуется прибегать к специальным средствам объединения блоков, принятым в других продуктах. Просто пользователь с помощью мыши рисует на экране линии, связывающие выходной соединитель одного блока с входным соединителем другого. Если связь недопустима, **Extend** не принимает ее. Для просмотра выходной информации, поступающей из блока, разработчик объединяет один из его выходных соединителей с блоком из библиотеки **Plotter**. Такие блоки обладают высокой степенью гибкости и могут принимать множество типов входных данных, выводя их в виде таблиц или диаграмм.

При построении модели центра обслуживания посредством **Extend** можно практически неограниченно менять частоту поступления заказов, время обслуживания одного клиента и другие параметры. Модель строится быстро и автоматически выводит результаты. **Extend** дает возможность проводить анализ чувствительности по многим переменным, участвующим в моделировании, и многократно исполнять модель, используя в качестве исходных данных различные значения переменных. Результаты каждого прогона выводятся на графопостроитель в различных цветовых режимах. Это позволяет разработчику исследовать модель при различных обстоятельствах и быстро принять решение. Пакет **Extend** снабжен функциями создания авторских моделей, с помощью

которых разработчик включает в окно модели текст, геометрические изображения и управляющие блоки, чтобы пользователи могли самостоятельно модифицировать модель. Для контроля за процессом моделирования и вывода результатов на дисплей применяется средство, организованное по принципу блокнота, которое удобнее в использовании, чем электронная таблица *Key Values* пакета **Process Charter**. Пакет **Extend** предоставляет подробное руководство пользователю, обучающую программу и примеры моделей из самых различных областей деятельности, которые могут служить основой для создания новых моделей, что, несомненно, облегчает процесс моделирования.

Недостатки пакета **Extend**:

- используется в полном объеме только на компьютерах типа **Macintosh**;
- высокая стоимость.

### 1.1.5. Пакет **Pilgrim**

Пакет **Pilgrim** поддерживает *дискретно-непрерывное* моделирование [3]. Он создан в России на базе Московского государственного университета экономики, статистики и информатики (МЭСИ) и нескольких компьютерных фирм.

Существует шесть основных понятий, на которых базируется концепция моделирующей системы.

1. **Граф модели**. Все процессы, независимо от количества уровней структурного анализа, объединяются в виде направленного графа.
2. **Транзакт** – это формальный запрос на какое-либо обслуживание. Он имеет набор динамически изменяющихся свойств и параметров. Пути миграции транзактов по графу стохастической сети определяются логикой функционирования компонентов модели в узлах сети.
3. **Узлы** графа сети представляют собой центры обслуживания транзактов. В узлах транзакты могут задерживаться, обслуживаться, породить семейства новых транзактов, уничтожать другие транзакты.

В **Pilgrim** имеется 17 типов узлов, которые предоставляют:

- возможность работы с непрерывными процессами;
- моделирование пространственной динамики;
- работу с ресурсами, представляющими собой деньги и материальные ценности, счета бухгалтерского учета, банковские счета.



4. **Событием** называется факт выхода из узла одного транзакта. События всегда происходят в определенные моменты времени. Они могут быть связаны и с точкой пространства.
5. **Ресурс** независимо от его природы в процессе моделирования может характеризоваться тремя параметрами: *мощностью*, остатком и *дефицитом*. Материальные ресурсы, например, подразделяются на две разновидности: *неперемещаемые* и *перемещаемые*. Неперемещаемый ресурс выделяется в определенном месте (как в реальности, так и в модели), это - конкретный служащий из обслуживающего персонала. Он не может перемещаться вместе с клиентом (транзактом). После обслуживания одного клиента он либо приступит к обслуживанию следующего, если есть очередь, либо будет отдыхать.
6. **Пространство** – географическое, декартова плоскость. Узлы, транзакты и ресурсы могут быть привязаны к точкам пространства и мигрировать в нем.

CASE – технология многослойного имитационного моделирования, которая в последние годы получает все большее распространение, используется в конструкторе моделей **Pilgrim**. Модель с точки зрения конструктора **Pilgrim** можно представить как набор следующих компонентов:

- граф модели;
- параметры инициализации модели;
- переменные модели;
- включенные в модель фрагменты программного кода на языке C++.

Основное достоинство конструктора **Pilgrim** заключается в том, что он позволяет:

- проводить многоуровневую (т.е. многослойную) иерархическую декомпозицию глобального процесса, разлагая его на составляющие компоненты (проводить структурный системный анализ);
- представлять каждый уровень структурной детализации в виде графического слоя;
- автоматически генерировать программный текст модели.

К достоинствам пакета **Pilgrim** следует отнести свойство коллективного управления процессом моделирования, невысокую стоимость, наличие интерфейсов с базами данных и создание пользовательских блоков с помощью языка программирования C++.

Недостатки пакета **Pilgrim**:

- сложная система обозначений;

- необходимо знать язык программирования C++.

#### 1.1.6. Пакет Vensim

Пакет **Vensim** предназначен для построения системно-динамических моделей, также как **Powersim** и **Ithink**. Он поддерживает *непрерывное* моделирование; разработан фирмой Ventana Systems.

Пакет **Vensim** обладает средствами оптимизации и статистики; возможностью создавать подключаемые к другим программам DLL-библиотеки; простой графического интерфейса, нацеленного на профессионалов; расширяемой библиотекой функций.

Недостатки пакета **Vensim**:

- нет возможности конвертирования данных;
- небольшое количество встроенных математических функций в версии PLE.

С помощью рассмотренных пакетов имитационного моделирования менеджеры получают возможность заглянуть в будущее и, например, эффективно укомплектовать штат предприятия, сделать лучшие инвестиции и успешно реорганизовать бизнес, не покидая при этом своего рабочего места.

## ГЛАВА 2. РАБОТА С ПРОГРАММНЫМ ПАКЕТОМ ITHINK

Существует множество аналогичных по своим целям и возможностям программных средств, которые могут быть использованы для решения тех или иных задач, возникающих в экономической практике. Одной из наиболее важных задач современной экономической практики является управление финансовыми потоками, циркулирующими на предприятии, в холдинге, между банком и его клиентами. Специфика «потокового» подхода к управлению заключается в планировании и контроле перемещений финансовых и материальных ресурсов, при этом в виде потока можно рассматривать не только движение денежных средств, но и функционирование предприятия в целом. Оперативное и эффективное управление этими потоками возможно только при наличии специализированных программных средств, например, таких, как системы имитационного моделирования, приведенные в главе 1, и экспертных систем.

Рассмотрим применение пакета **Ithink** – уникального средства имитационного моделирования производственных и финансовых проектов и процессов [7-12]. Пакет **Ithink** предназначен для следующих наиболее важных групп пользователей:

а) инвестиционных компаний, брокеров, дилеров ценных бумаг (эта группа пользователей с помощью **Ithink** осуществляет планирование инвестиционных операций, прогнозирование рыночной конъюнктуры и доходности вложений);

б) аналитических отделов банков и финансово-промышленных групп (в аналитических отделах банков – самая широкая сфера применения имитационных моделей: прогнозирование, «обкатка» структурных схем и инвестиционных проектов, выбор и

обоснование оптимальной стратегии. **Ithink** – это экспертная программа, пригодная для решения задач планирования и управления финансово-промышленной группой.);

в) отделов проектных исследований банка (пакет **Ithink** способен обеспечить достаточно глубокую детализацию проектных документов, например, плановые графики и таблицы могут быть сделаны в нужном временном масштабе);

г) консультационных и проектных компаний (пакет **Ithink** позволяет наглядно продемонстрировать суть рекомендаций и последствия их реализации. Так, схема автоматизации бухгалтерии и управленческого учета может быть предварительно просчитана на имитационной модели);

д) региональных органов власти (с помощью пакета **Ithink** возможно моделирование региональных экономик: планирование хозяйственных систем областного, городского и муниципального масштаба. Пакет обеспечивает повышение эффективности управления процессами снабжения и распределения ресурсов. Возможно моделирование сезонных циклов, что актуально для сельского хозяйства. **Ithink** может применяться для планирования социальной сферы и медицинских услуг. Актуальной задачей может быть прогнозирование налоговых сборов и управление местными бюджетами);

е) «отраслевых» областей (пакет **Ithink** может применяться для управленческого моделирования объектов топливно-энергетического комплекса, металлургических, химических и других предприятий, включая предприятия с непрерывным производственным циклом).

Следует отметить следующие достоинства пакета структурного моделирования **Ithink**:

- 1). *Простота в работе.* Работа с пакетом **Ithink** не требует специальных навыков и знания языка программирования. Модель разрабатывается путем размещения на рабочем листе встроенных строительных блоков, соединенных стрелками. Стрелки указывают направление финансовых и материальных потоков, а также отображают структурные взаимосвязи модели.
- 2). *Уникальный эффект визуализации.* Автор модели постоянно видит всю систему целиком. При любом изменении схемы модели автоматически изменяются алгоритм и программный код, который генерируется автоматически.
- 3). *Отсутствие «жесткой привязки» к какому-либо кругу задач.*
- 4). *Развитые средства анализа чувствительности, обеспечивающие автоматическое многократное исполнение модели с различными входными данными.*
- 5). *Поддержка множества форматов входных данных.*

В процессе моделирования в окне модели формируется структурная схема модели из встроенных строительных блоков. Затем между ними устанавливаются взаимосвязи и

генерируется программный код. Оператору остается только ввести функциональные зависимости и числовые параметры, после этого модель готова к запуску. Основных строительных блоков пять: **фонд**, **поток**, **конвертер**, **коннектор**, **ромбоидальное определение процесса**. **Фонд** – количество чего-либо, существующее в данный момент времени и измеряемое либо в денежных, либо в физических единицах (2 тысячи рублей, 5т макарон, 200 рейтинговых баллов и т.д.). Фонд в **Ithink** изображается прямоугольником, который способен накапливать, аккумулировать единицы фонда. По умолчанию фонд представляет собой резервуар, но существуют еще три типа фондов: конвейер, очередь и печь. **Поток** - это процесс, протекающий непрерывно во времени, оценить который можно в физических или денежных единицах, соотнесенных с каким-либо временным интервалом (рубли в месяц, литры в час, стоимость акций на время закрытия биржи в данный день и т.д.). В **Ithink** поток изображается фигурой, состоящей из путепровода, вентиля, регулятора потока и указателя направления. По характеру использования, потоки подразделяются на ограниченные и неограниченные, однонаправленные и двунаправленные, конвертируемые и неконвертируемые. **Конвертеры** в **Ithink** – преобразователи модельных единиц, которые изображаются окружностями. Они могут содержать значения констант или внешних входных переменных, подсчитывать значения алгебраических выражений или использоваться для хранения графических функций. **Коннектор** предназначен для связи между собой элементов модели. **Ромбоидальное определение процесса**– это механизм для управления запутанными схемами, связанный с представлением определения процессов внутри модели. С помощью этого блока можно скрыть сложность определенных операций.

## **2.1. Основные операции**

### **2.1.1. Уровни представления моделей**

В **Ithink** модели представляются тремя иерархическими уровнями: уровень интерфейса (высокоуровневое представление блок-схемой), уровень модели, уровень программного кода. Рассмотрим эти уровни подробнее.

#### **2.1.1.1. Уровень интерфейса**

На данном уровне модель представляется в виде блок-схемы. На Рис.2.1 мы видим окно модели.

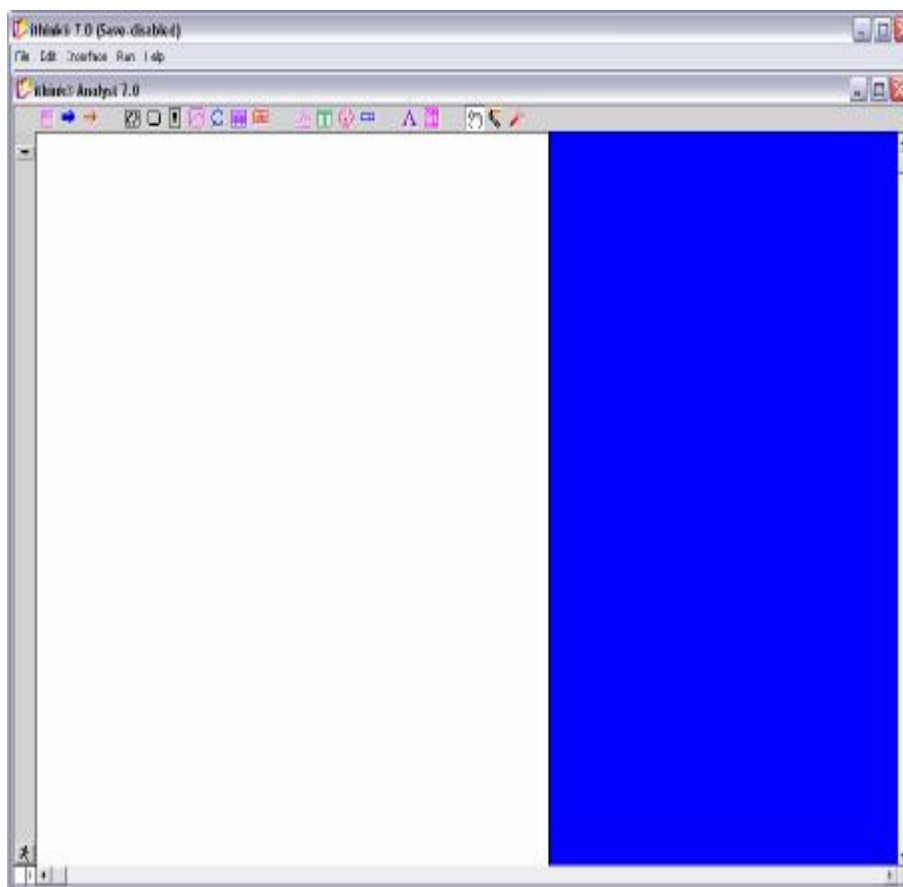


Рис.2.1

В верхней части окна находятся **панель меню** и **панель инструментов**. На панели инструментов находятся кнопки строительных блоков (**фрейм**, **поток**, **связь**, **коннекторная связь**), кнопки различных объектов и инструментов, используемых в **Ithink** для создания и управления моделями. **Белый фон** – это **рабочая поверхность**, ее размер по умолчанию соответствует формату А4. Но в случае, если необходимо занять все пространство, можно просто поместить элемент на **синий фон**, и тогда размер рабочей площади автоматически увеличивается. Нажатие на кнопки с изображениями **плюса** или **минуса**, находящиеся в левом нижнем углу окна, позволяет увеличить или уменьшить в два раза масштаб изображения. С левой стороны от рабочего листа размещена **кнопка с изображением стрелки**. Если нажать на нее с помощью левой кнопки мыши, то можно перейти на модельный уровень. **Кнопка с изображением бегущего человечка** аналогична пункту меню **Run**, и вызывает появление окна управления запуском. Это окно имеет вид, представленный на Рис. 2.2.



Рис.2.2

В этом окне расположены следующие кнопки (слева на право): **Запуск, Пауза, Стоп, Параметры имитации.**

### 2.1.1.2. Уровень модели

Уровень модели - базовый уровень, на котором строится модель при помощи потоковых схем. На Рис.2.3 показано окно программы **Ithink** на модельном уровне.

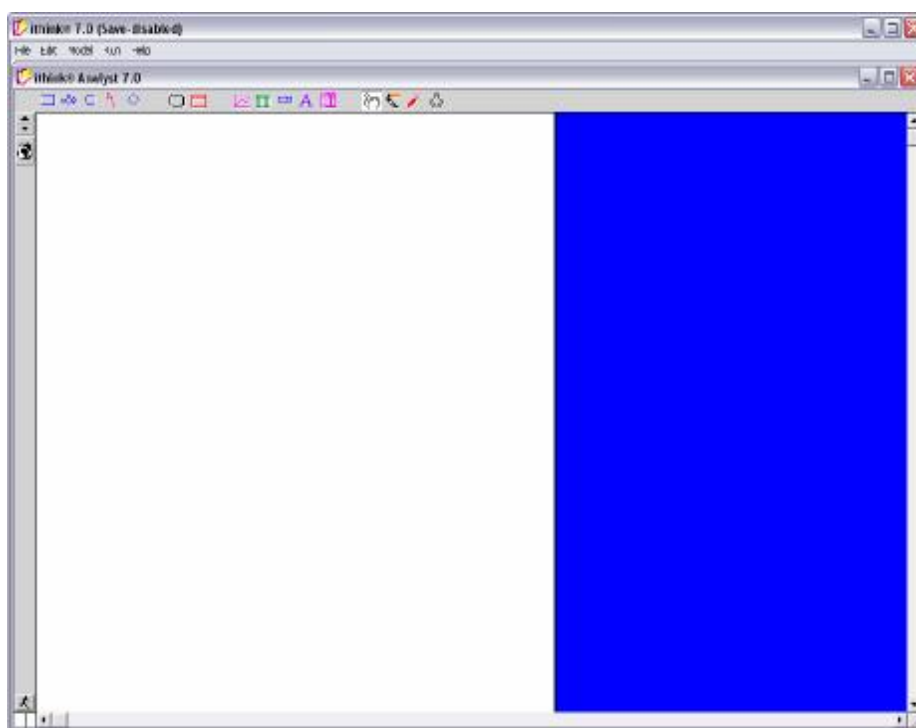


Рис.2.3

На панели инструментов находятся **кнопки строительных блоков, инструментов и объектов** (все они будут описаны ниже). Кнопки с изображением стрелок позволяют перемещаться на другие уровни: **стрелка вверх** – на уровень интерфейса, **стрелка вниз** – на уровень программного кода. Кнопка, которая находится ниже, может иметь следующие варианты внешнего вида:



- В этом случае вы находитесь на этапе составления схемы модели. При открытии строительных блоков диалоговое окно будет содержать только опции.



- В этом случае вы находитесь на этапе числового описания модели, на котором вы можете задать числовые значения и функциональные зависимости между элементами модели. В случае, если значение блока не задано, то на нем появляется изображение знака вопроса.

### 2.1.1.3. Уровень программного кода

В результате создания модели на модельном уровне программный код на этом уровне генерируется автоматически. Окно программы имеет следующий вид:

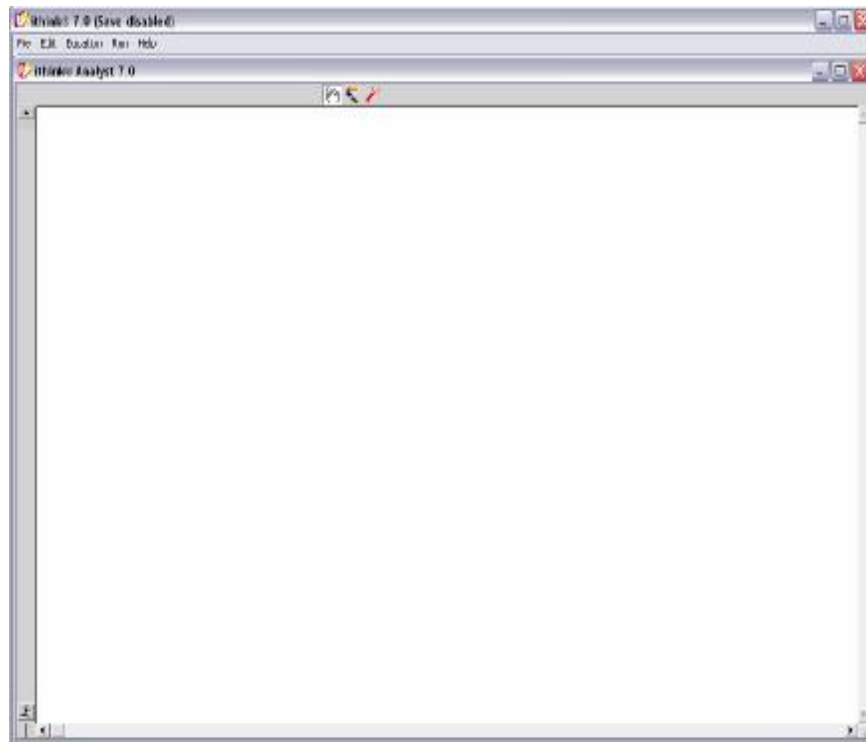


Рис.2.4

## 2.2. Меню

### 2.2.1. Меню File

Многие из пунктов, входящих в пункт меню **File**, являются стандартными для **Windows** (Рис.2.5).



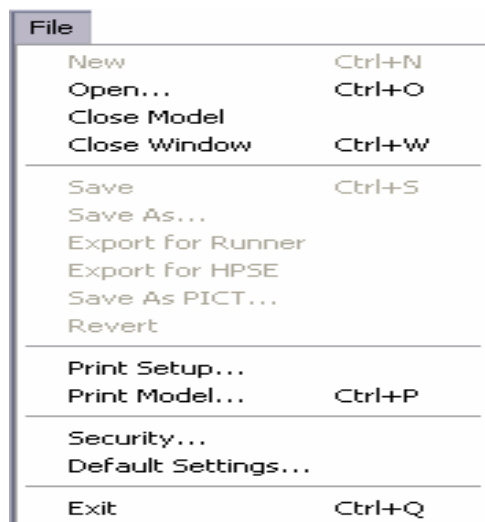


Рис.2.5

**New** – создает новую модель. Так как программа **Ithink** позволяет работать только с одной моделью, то данный пункт меню является неактивным, если вы работаете в ранее созданном файле. Для того чтобы создать новый файл, необходимо закрыть старую модель.

**Open** – вызывает стандартный диалог для открытия файла.

**Close Model** – закрывает открытую ранее модель. Если она отличается от той, которая сохранена на диске, то будет предложено сохранить изменения.

**Close Window** – закрывает активное окно (например, график или таблицу). Если используется для одного из трех уровней программного пакета (уровни интерфейса, модели и программного кода), то данный пункт меню эквивалентен **Close Model**.

**Save** – сохраняет модель в текущем состоянии на диске в файл с расширением **.itm**.

**Save As** – вызывает стандартное диалоговое окно для сохранения модели под новым именем в выбранной директории или папке.

**Export for Runner** – сохраняет модель в специальном формате, который позволяет ей существовать как отдельное приложение. Автор модели может контролировать доступ к различным компонентам и интерфейсу модели.

**Export for NetSim** – сохраняет файл модели в виде, который распознается программой **NetSim Creator**.

**Export Text Boxes** - позволяет скопировать содержание выбранных текстовых блоков в буфер или сохранить их в текстовом файле.

**Save As PICT or Text** – позволяет сохранить данные в формате **.pct** или **.txt** (в зависимости от того, какое окно является активным).

**Revert** – возвращает модель в то состояние, в котором она была последний раз сохранена. Все изменения, сделанные после сохранения, будут утеряны.

**Print Setup or Page Setup** – вызывает диалоговое окно, в котором задаются параметры печати или страницы.

**Print** – вызывает диалоговое окно для печати. Его вид зависит от того, какой уровень является активным, а также от того, что выбрано внутри уровня.

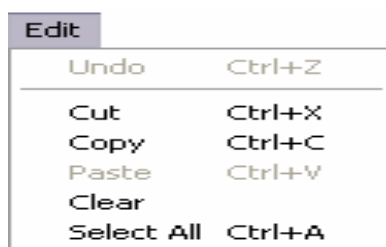
**Security** – позволяет ограничить доступ к модели.

**Default Settings** – позволяет определить настройки интерфейса и моделирования, которые будут использоваться по умолчанию.

**Exit** – закрывает приложение. Если изменения в модели не были сохранены, то появится диалоговое окно с предложением сохранить изменения.

### 2.2.2. Меню Edit

Окно пункта меню Edit имеет вид (Рис.2.6):



Edit	
Undo	Ctrl+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Clear	
Select All	Ctrl+A

Рис.2.6

**Undo** – отмена последнего действия.

**Cut** – вырезает выбранную информацию из окна в буфер.

**Copy** – копирует выбранную в модели информацию в буфер.

**Paste** – вставляет информацию из буфера в модель. Однако следует отметить, что данные должны быть согласованными (нельзя вставить картинку на уровень программного кода).

**Clear** – удаляет выбранные элементы модели с уровня интерфейса или модельного уровня, выбранные переменные из графика или таблицы. Данная команда эквивалентна нажатию клавиши **backspace**.

### 2.2.3. Меню Interface, Model and Equation

Название данного меню зависит от того, на каком уровне модели вы находитесь. Необходимо отметить, что большинство пунктов из этих меню совпадают.



Рис.2.7

**Find** – этот пункт меню имеется на всех трех уровнях представления модели. После выбора этого пункта появляется диалоговое окно со списком всех блоков и подпрограмм. После выбора одного из пунктов списка искомый объект в окне соответствующего уровня будет выделен цветом.

**Find Next** – данный пункт меню доступен на модельном уровне, если вы выбрали переменную, которая скопирована при помощи инструмента **Фантом**. После выбора этого пункта вы получите все копии в порядке создания.

**Find Controller** – этот пункт меню доступен на уровне модели, если вы выбрали переменную, которая задается с помощью объектов (например, переключатель, ручка или слайдер) на уровне интерфейса. Около переменной появляется маленькая иконка с изображением соответствующего объекта.

**Align to Grid** - позволяет на уровне интерфейса или модели выровнять выделенные объекты по невидимой сетке. Размер сетки можно определить в диалоговом окне настроек модели.

**Open Selection** - данное меню зависит от того, на каком уровне вы находитесь, и какой объект выбран внутри окна. Этот пункт эквивалентен двойному щелчку мыши по объекту. С помощью **Open Selection** можно открыть выбранный блок, определить график, таблицу или цикл.

**Show / Hide Pads** – позволяет показать или скрыть окна графиков, таблиц или циклов.

**Open / Close Decision Diamonds** – данный пункт меню доступен только на уровне модели. Когда в модели присутствуют блоки ромбоидального определения процесса или подмодели, то выбор этого пункта позволяет открыть их пространство для редактирования или скрыть его под иконкой.

**Clear Second Position.** Вторичное местоположение – это расположение модельных строительных блоков (фонды, потоки, конверторы, коннекторы), если пространства ромбоидальных определений процессов или подмоделей открыты. Данная команда позволяет очистить все данные, которые имеют вторичное местоположение.

**Hide** – этот пункт меню позволяет выборочно скрыть отображенные элементы на уровне интерфейса или модели. Данную команду можно использовать для конверторов, фантомов, коннекторов, коннекторных связей на уровне интерфейса, иконок графиков и таблиц, кнопок.

**Show** – позволяет вновь увидеть те элементы, которые были скрыты ранее.

**Restore** – этот пункт меню восстанавливает первоначальное значение для следующих объектов: Таблица, График, Числовой дисплей, Слайдер, Ручка, Переключатель и т.д.

**Interface, Model & Equation Preferences** – эти пункты меню вызывают настройки для соответствующих уровней.

ü **Interface Prefs** (Рис.2.7) – вызывает диалоговое окно для установки значений видимых настроек уровня интерфейса (Рис.2.8).

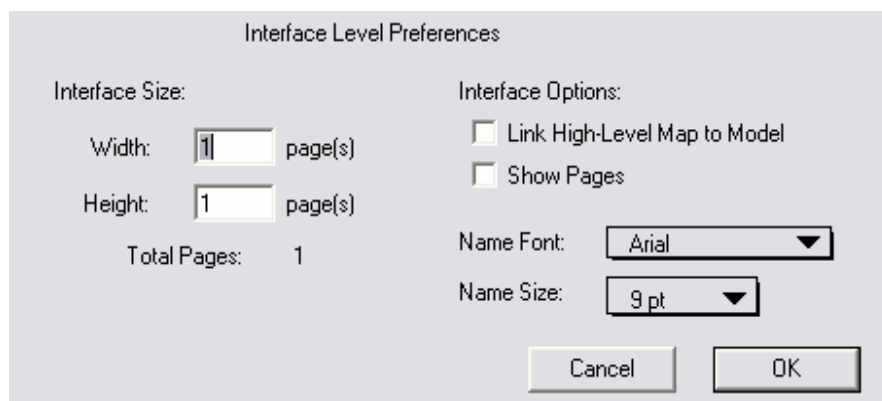


Рис.2.8

При задании чисел в графах **Width** и **Height**, изменяется размер уровня интерфейса, который измеряется в страницах.

Флажок **Link High-Level Map to Model** позволяет связать объекты блок-схемы на уровне модели и элементы модели на модельном уровне. Если флажок поставлен, то при

создании модели на модельном уровне автоматически формируется схема на уровне интерфейса (фрейм сектора, коннекторы и потоки связываются с фреймом, коннекторной связью и потоковой связью).

Флажок **Show Pages** делает видимыми все страницы и их границы.

Иконка **Model Prefs** (Рис. 2.7) - вызывает диалоговое окно для установки значений видимых настроек модельного уровня (Рис. 2.9).

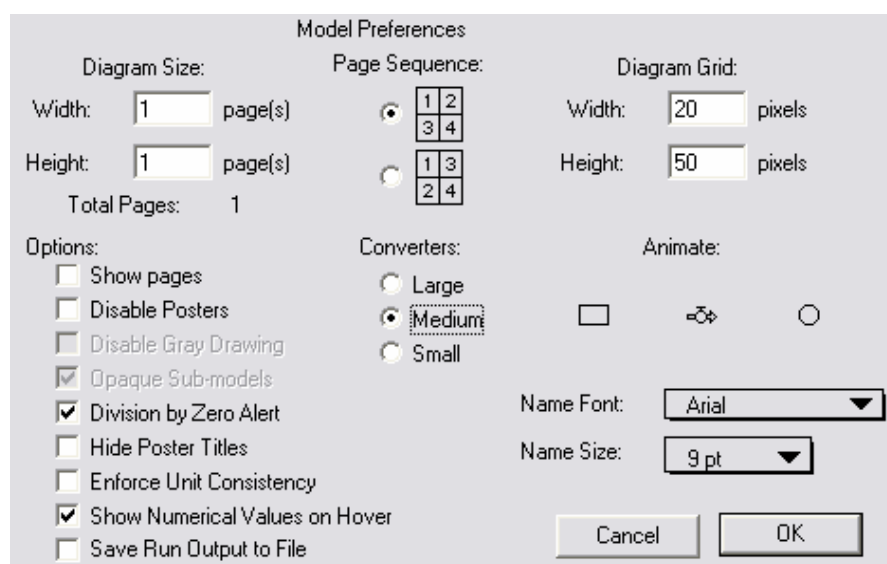


Рис.2.9

Задавая числа в графах **Width** и **Height** под заголовком **Diagram Size**, изменяется размер модельного уровня, который измеряется в страницах.

Переключатель **Page Sequence** позволяет выбрать порядок расположения страниц на модельном уровне.

В полях **Width** и **Height** в разделе **Diagram Grid** можно задать размеры невидимой сетки в пикселях, которые используются для выполнения команды **Align to Grid**.

Переключатели **Converters** позволяют выбрать большой, средний или маленький размер конверторов.

Чтобы задать анимацию фондов, потоков или конверторов надо щелкнуть мышкой по соответствующим кнопкам. Вокруг выбранных элементов появится рамка. Можно выбирать как один, так и несколько элементов.

В раскрывающихся списках **Name Font** и **Name Size** задаются название и размер шрифта.

Теперь рассмотрим свойства, выбор которых совершается путем установки соответствующего флажка. Флажок **Show Pages** делает видимыми все страницы и их границы. Флажок **Disable Posters** отключает информационные сообщения, созданные внутри модели. В случае если в модели присутствуют подмодели, то выбор флажка **Disable Gray Drawing** означает, что основной уровень строительных блоков не будет нарисован серым цветом, когда одна или несколько подмоделей открыты. Если флажок **Opaque Sub-models** включен, то фон подмодели будет непрозрачным, в противном случае он будет прозрачным. Флажок **Division by Zero Alert** обуславливает появление сообщений о делении на ноль или установления идентичности вызывающих модельных переменных; если он отключен, то модель продолжает генерировать данные даже в случае деления на ноль. В случае включения флажка **Hide Poster Titles** все заголовки сообщений будут скрыты. Флажок **Enforce Unit Consistency** означает, что в случае если единицы измерения в вашей модели не согласованы, после нажатия клавиши **OK** для выхода из диалогового окна задания уравнений, на соответствующем строительном блоке появится изображение восклицательного знака (!). Несоответствие единиц измерения необходимо устранить до запуска модели.

Флажок **Show Numerical Values on Hover** активизирует отображение функциональных сведений об элементах модели. То есть, если вы подведете курсор к одному из блоков модели, то рядом с ним появится числовое значение на последний момент времени и маленький график. Если флажок отключен, то такой подсказки появляться не будет. Если отмечен флажок **Save Run Output to File**, то все выходящие переменные модели в течение всех временных периодов будут сохранены в файл, который может быть открыт с помощью других приложений (например, **Excel**).

Ü **Equation Prefs** (Рис. 2.7) - вызывает диалоговое окно для установки значений видимых настроек уровня программного кода (Рис. 2.10).

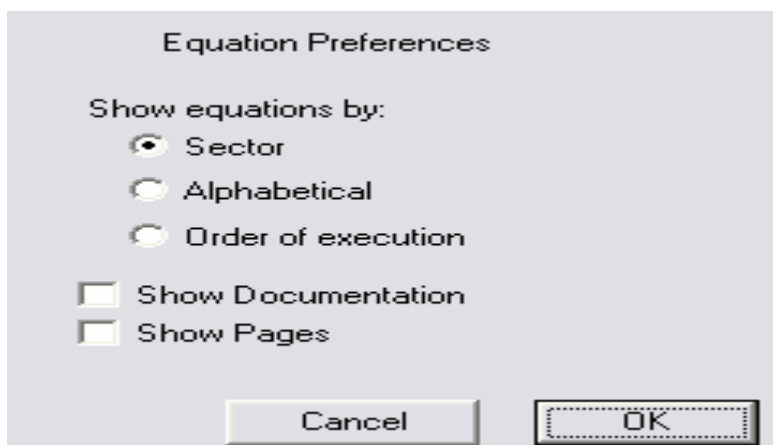


Рис.2.10

Переключатель задает порядок, в котором уравнения могут быть напечатаны или сохранены. Они могут быть расположены по секторам, в алфавитном порядке или в порядке выполнения.

Флажок **Show Documentation** позволяет показать некоторую документацию как часть списка уравнений.

Флажок **Show Pages** делает видимыми все страницы и их границы, и позволяет их напечатать при печати уравнений.

**Edit Links** – позволяет редактировать связи модели с другими приложениями, например, с **Excel** (Рис.2.7).

**Array Editor** – используется для задания размерностей и элементов, которые будут использоваться для создания массивов переменных (Рис.2.7).

**Unit Editor** – содержит список всех доступных в вашей модели единиц измерения и позволяет создать новые (Рис.2.7).

#### 2.2.4. Меню Run

Содержание этого меню эквивалентно панели управления запуском (Рис.2.11).



Рис.2.11

**Run** – запускает имитационную модель в соответствии с параметрами, заданными в **Sector Specs** и **Sensi Specs**.

**Pause** – делает паузу в работе модели.

**Stop** – останавливает симуляцию модели.

**Sector Specs** – данный пункт меню позволяет выбрать те сектора, которые вы хотите запустить. Внешний вид диалогового окна показан на Рис.2.12.

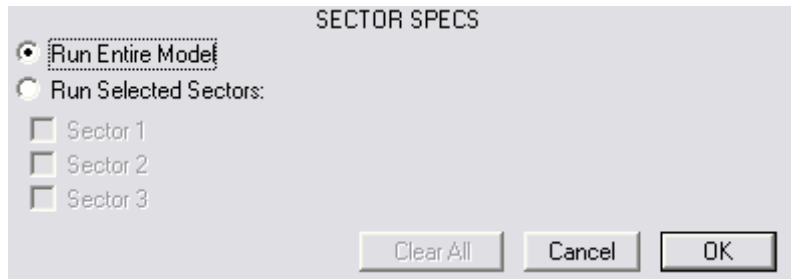


Рис.2.12

Как показано на Рис.2.12, в диалоговом окне присутствует два переключателя. Первый переключатель **Run Entire Model** подразумевает, что вы хотите запустить все элементы модели. Второй переключатель **Run Selected Sectors** активизирует флажки, выбор которых обеспечивает запуск только отдельных секторов. Кнопка **Clear All** очищает все отмеченные флажки.

**Sensi Specs** – этот пункт меню (Рис.2.11) позволяет задавать характеристики установок чувствительности для запуска и делает доступным анализ чувствительности модели.

На Рис.2.13 представлено диалоговое окно, в котором можно установить характеристики для анализа чувствительности модели.

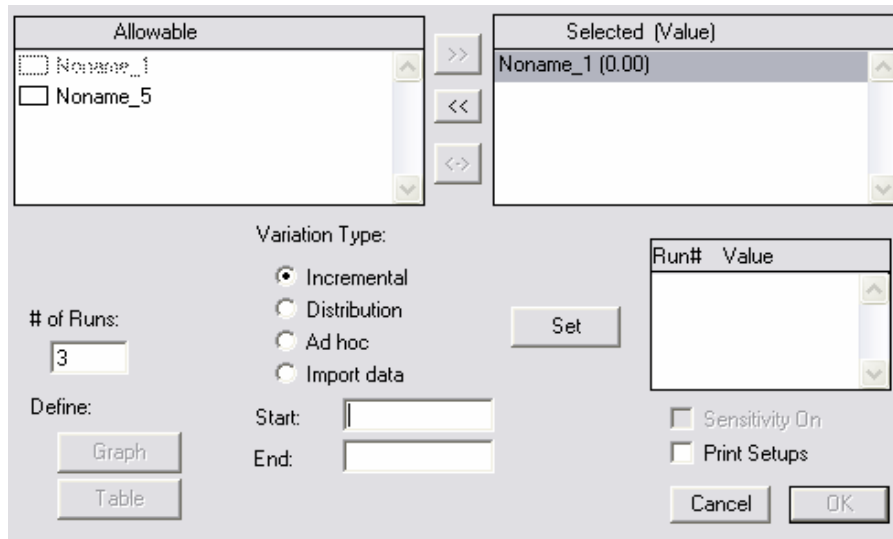


Рис. 2.13

Выберите из списка **Allowable** переменные (в списке указаны те блоки, которые имеют постоянные значения), для этого щелкните мышкой по элементу и нажмите на кнопку с изображением >>.



В поле **# of Runs** указывается число запусков, которые вы хотите сделать при анализе чувствительности. Надо отметить, что это поле недоступно, если включен переключатель **Import data**.

С помощью переключателей **Variation Type** задается тип вариации. Если выбран пункт **Incremental**, то появляются поля **Start** и **End**, в которых указываются начальное и конечное значения анализа чувствительности соответственно.

После нажатия на кнопку **Set** в списке **Run# Value** появятся значения параметра, для которых будет происходить имитация.

Если выбран пункт **Ad hoc**, то каждое значение для имитации вы вводите самостоятельно в поле **Ad hoc Values**. После ввода каждого значения надо нажимать кнопку **Set**.

При выборе пункта **Distribution** задается вид распределения путем нажатия на кнопку, которая появляется рядом с переключателем. Это может быть нормальное распределение, и тогда задаются значения средней величины, стандартного отклонения и случайной величины, или равномерное распределение, и тогда надо задать минимальное и максимальное значения, а также случайную величину. Чтобы выбрать пункт **Import data** необходимо скопировать некоторые числовые данные в буфер. После выбора этого переключателя справа от него появится кнопка **Import**. После нажатия на нее в списке **Run# Value** появятся значения из буфера. Число запусков устанавливается автоматически в зависимости от количества данных в буфере.

Кнопки **Graph** и **Table** позволяют перейти к окну графика или таблицы соответственно.

Флажок **Sensitivity On** включает чувствительность.

Флажок **Print Setups** позволяет напечатать структуру анализа чувствительности вместе со сравнительными графиками и таблицами.

**Run Specs** – этот пункт меню (Рис.2.11) вызывает появление диалогового окна, в котором можно задать продолжительность моделирования, временной шаг между вычислениями (**DT**), интервал между перерывами в симулировании, метод итерации, единицы измерения времени в модели и тип запуска (в нормальном или циклическом времени). Данное диалоговое окно представлено на Рис.2.14.

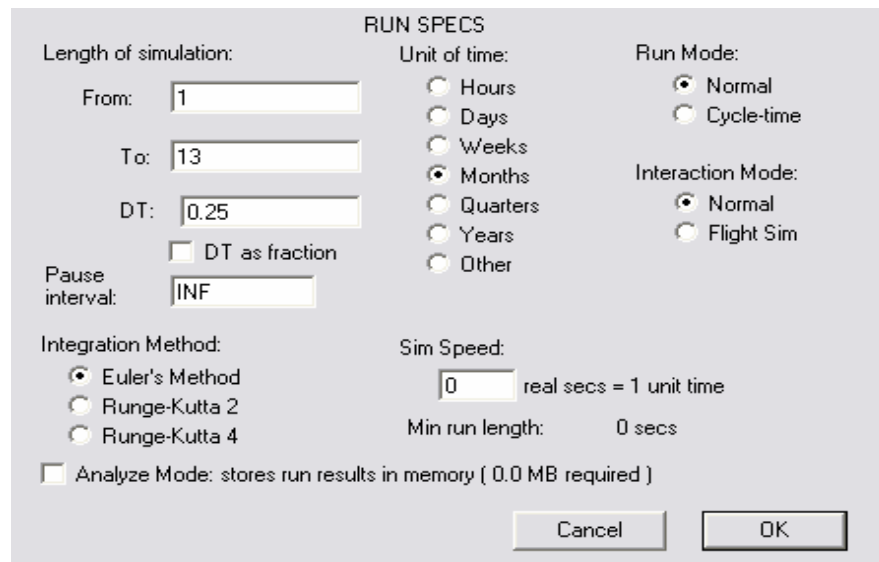


Рис.2.14

В полях **From** и **To** указываются начало и конец времени моделирования. Значение второго поля должно быть больше. В поле **DT** вводится значение временного шага симуляции – это промежуток времени, через который производятся вычисления. В поле **Pause interval** по умолчанию указано значение бесконечность. Если интервал между паузами будет меньше продолжительности моделирования, то по прошествии этого времени симулирование будет остановлено до тех пор, пока не будет нажата кнопка **Start** на панели управления запуском. При помощи переключателей выбирают единицы измерения времени (часы, дни, недели, месяцы, кварталы, годы или какие-либо другие единицы); метод итерации (метод Эйлера, метод Рунге-Кутта 2 или метод Рунге-Кутта 4); тип запуска (в нормальном или циклическом времени) и тип взаимодействия (обычный или во времени). В поле **Sim Speed** указывается количество секунд в единице модельного времени, в случае если выбран переключатель **Flight Sim**. Флажок **Analyze Mode** позволяет сохранить результаты запуска в памяти. В этом случае в модели появится маленький график с предыдущими результатами, если вы изменяете значения переменных.

**Range Specs** (Рис.2.11) – вызывает появление диалогового окна (Рис.2.15), в котором дан список всех модельных переменных с указанием их последних глобальных максимальных и минимальных значений.

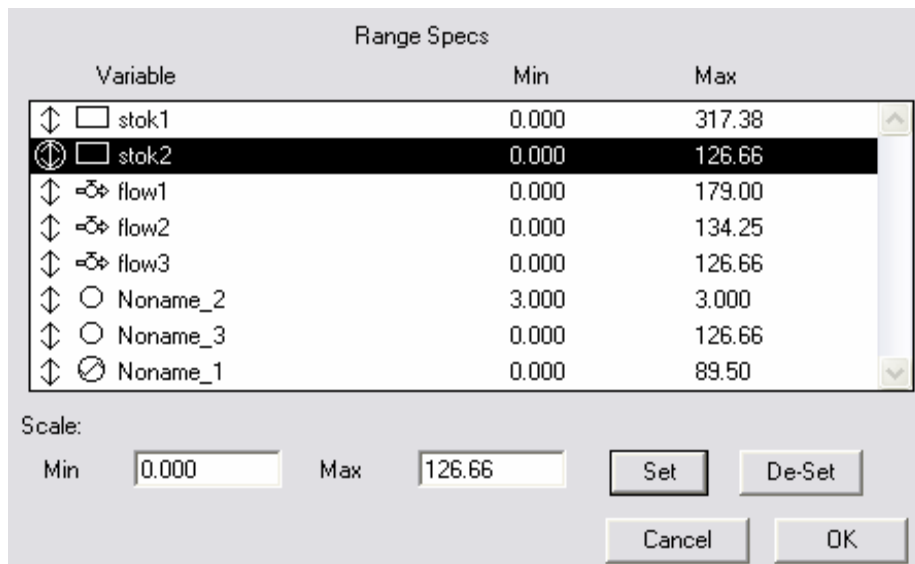


Рис.2.15

С помощью кнопок устанавливается или убирается глобальный масштаб переменных. Надо также отметить, что значения глобального максимума и минимума могут быть установлены при описании графика.

**Check Units** (Рис. 2.11) – позволяет проверить соответствие единиц измерения в модели, то есть если единицы измерения будут заданы неверно, то строительный блок с неправильными единицами будет выделен.

### 2.2.5. Меню Help

Окно меню **Help** представлено на Рис.2.16.

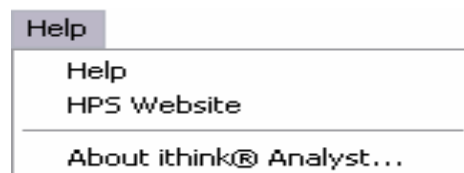


Рис.2.16

**Help** – вызывает файл справки.

**HPS Website** – позволяет перейти на официальный сайт компании **High Performance Systems** [www.hps-inc.com](http://www.hps-inc.com).

**About ITHINK / STELLA** вызывает появление окна с информацией о серийном номере, количестве памяти, номером версии и т.д.

## 2.3. Основные строительные блоки


Всего в программе **Ithink** существует восемь строительных блоков, из них три на уровне интерфейса и пять на модельном уровне.

### 2.3.1. Строительные блоки на уровне интерфейса

Существует два способа создания строительных блоков на уровне интерфейса:

1. Если в меню **Interface** вы выберете пункт **Interface Prefs** и отметите флажком пункт **Link High-Level Map to Model**, то при создании модели на модельном уровне и объединении некоторых элементов фреймом сектора на уровне интерфейса блок-схема сформируется автоматически.
2. Построение блок-схемы непосредственно на уровне интерфейса при помощи выбора необходимых блоков на панели инструментов и перемещения их на рабочий лист.

#### 2.3.1.1. Фрейм

Этот блок позволяет представить высший уровень иерархии модели, что упрощает понимание структурных связей и взаимодействия отдельных групп элементов модели. Также облегчается управление связанным с ним фреймом сектора на модельном уровне. Чтобы создать фрейм, надо нажать на кнопку , а потом щелкнуть в том месте рабочего листа, куда вы хотите его поместить. Внешний вид фрейма представлен на Рис.2.17.

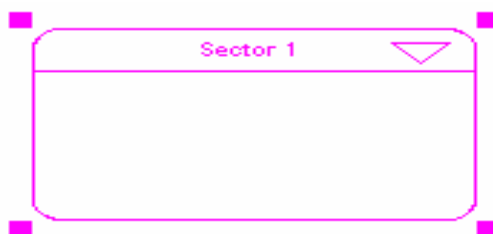


Рис.2.17

Чтобы активизировать фрейм, надо щелкнуть по нему левой кнопкой мыши, вокруг этого фрейма появятся квадратики, с помощью которых можно изменять его размер. В верхней строке находится название соответствующего сектора. Стрелка в правом верхнем углу позволяет перейти на модельный уровень к соответствующему фрейму сектора. Двойной щелчок левой кнопкой мыши на фрейме вызывает диалоговое окно описания фрейма (Рис.2.18).

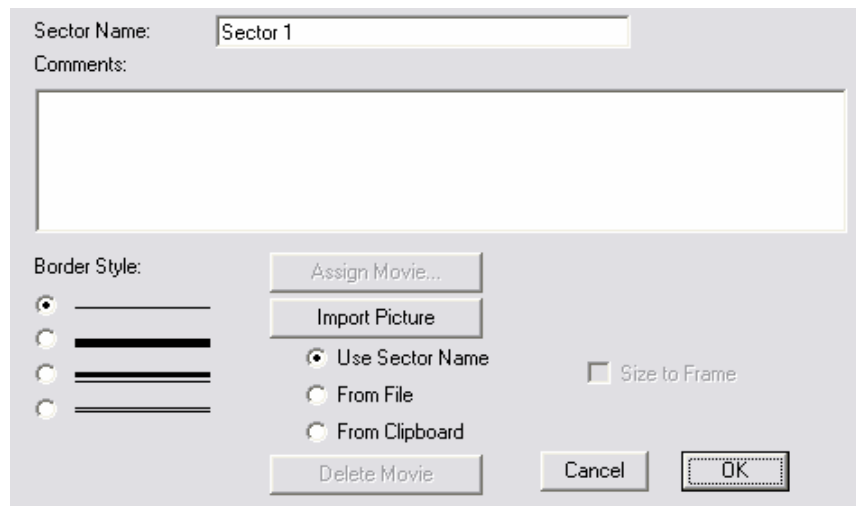



Рис.2.18

В поле **Sector Name** задается имя сектора (при его изменении новое название появляется и в заголовке фрейма сектора на модельном уровне), в поле **Comments** можно написать заметки и комментарии. С помощью переключателя **Border Style** можно выбрать внешний вид рамки. Кнопка **Assign Move** доступна, когда строительные блоки находятся в связанных секторах, и компьютер поддерживает **Quick Time**. Она эквивалентна назначению анимации внутри фрейма сектора.

Кнопка **Import Picture** позволяет поместить некоторый объект внутри фрейма. Это может быть имя сектора, картинка из файла или из буфера. Осуществить выбор можно с помощью соответствующего переключателя. Кнопка **Delete Movie** удаляет назначенную анимацию или картинку из фрейма. Флажок **Size to Frame** позволяет полностью поместить картинку внутри фрейма.

#### 2.3.1.2. Потокосвязь

**Потоковые связи** позволяют представить материальные потоки между процессами в вашей модели (Рис.2.19). Чтобы создать потокосвязь, надо нажать на кнопку . После этого надо щелкнуть левой кнопкой мыши по тому фрейму, из которого выходит потокосвязь, и, не отпуская левой кнопки, подвести связь к нужному фрейму. Данные действия необходимо осуществлять лишь при использовании второго способа создания блок-схем на уровне интерфейса.

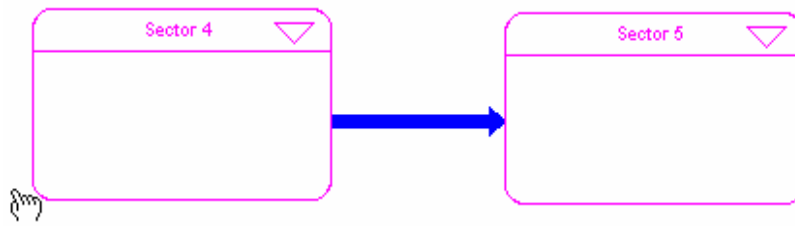


Рис.2.19

Следует отметить следующие особенности потоковой связи:

- Если вы хотите изогнуть трубку потока, то в месте изгиба надо нажать клавишу **Shift**, не отпуская при этом левую кнопку мыши.
- На уровне интерфейса вы можете иметь максимум одну потоковую связь из фрейма А в фрейм В и максимум одну потоковую связь в обратном направлении.

Управление потоковой связью происходит аналогично управлению фреймом. После двойного щелчка левой кнопкой мыши на потоковой связи появляется диалоговое окно описания потоковой связи (Рис.2.20).

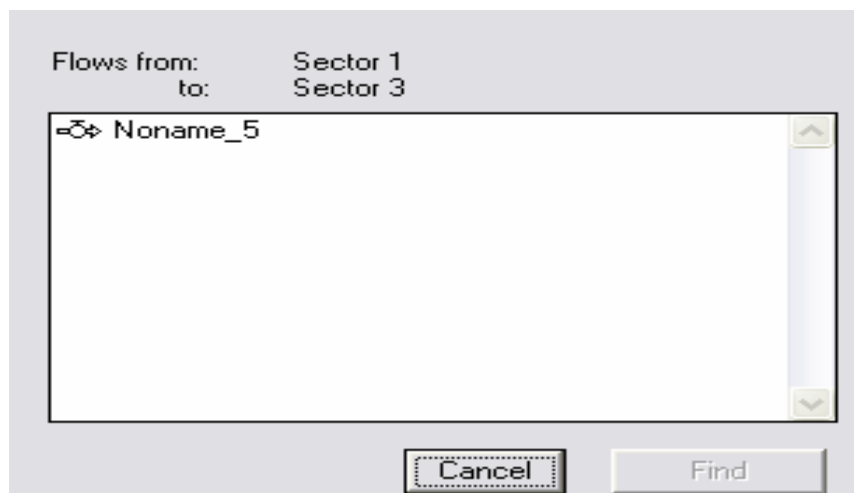



Рис.2.20

В диалоговом окне (Рис.2.20) указывается начало и конец потоковой связи, а также все межсекторные потоки на модельном уровне, которые на уровне интерфейса представляются агрегированной потоковой связью. Данный список будет активен только в том случае, если стоит флажок **Link High-Level Map to Model**. Если выбрать один из потоков, указанных в списке, и нажать на кнопку **Find**, то программа перейдет на модельный уровень и выделит искомый поток.

### 2.3.1.3. Коннекторная связь

Подобно остальным блокам, **коннекторная связь** позволяет представить все коннекторы, содержащиеся в модели, в общем виде. Если флажок **Link High-Level Map to Model** не стоит, то для создания коннекторной связи нажмите на кнопку , далее действия совпадают с действиями при создании потоковой связи. Коннекторная связь имеет вид тонкой красной стрелки (рис.2.21).

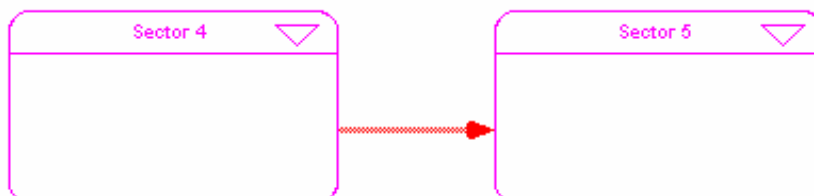


Рис.2.21

Диалоговое окно определения коннекторной связи представлено на Рис.2.22.

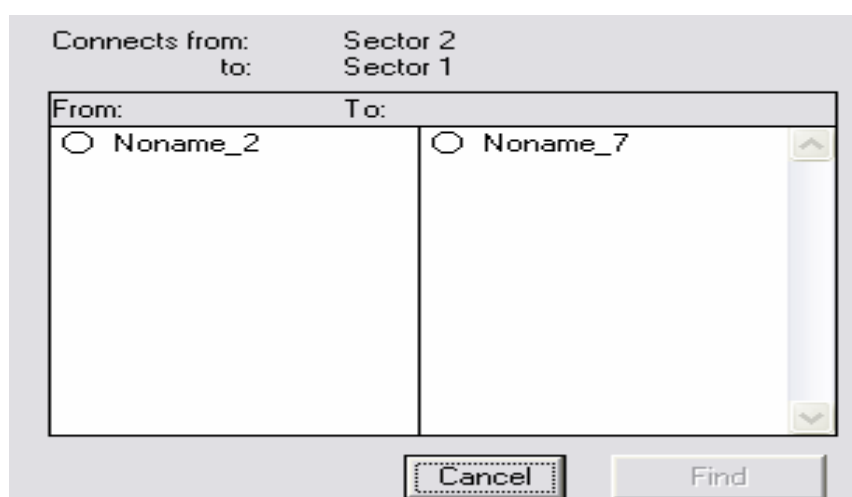


Рис.2.22

В списке указаны все коннекторы уровня модели, которые на уровне интерфейса агрегируются этой коннекторной связью.

### 2.3.2. Строительные блоки на уровне модели

Основных типовых структурных блоков немного - всего пять. Однако в процессе работы выясняется, что этого вполне достаточно, поскольку они весьма универсальны. Строительные блоки изначально наделены определенными свойствами. Они зафиксированы в соответствующем фрагменте программы модели. Взаимодействие между

структурными блоками обеспечивается при помощи стрелок, которые проводятся мышью. Стрелки указывают направления взаимосвязей и передачи данных между блоками. Характер взаимосвязей вводится в виде формул. Адресация “входа” и “выхода” подпрограмм блоков формируется автоматически в соответствии со стрелками (коннекторами), проведенными оператором на структурной схеме.

Типовыми структурными элементами являются: *фонд*, *поток*, *конвертер*, *коннектор*, *ромбоидальное определение процесса*. Рассмотрим их подробнее.

### 2.3.2.1. Фонды

**Фонд** – количество чего-либо, существующее в данный момент времени и измеряемое либо в денежных, либо в физических единицах (2 тысячи рублей, 5т макарон, 200 рейтинговых баллов и т.д.). Фонд в **Ithink** изображается прямоугольником, который способен накапливать, аккумулировать единицы фонда. Они пополняются через потоки входные и растрачиваются через потоки выходные. По умолчанию фонд представляет собой **резервуар**, но существуют еще три типа фондов: **конвейер**, **очередь** и **печь**.

### 2.3.2.2. Резервуар

Суть **резервуара** в том, что он просто суммирует все входящие потоки и вычитает все выходящие. Единицы измерения потоков для него эквивалентны, и разницы между потоками данный блок не делает.


Для того чтобы создать данный блок, необходимо нажать на кнопку  и потом щелкнуть в том месте рабочего листа, в котором вы хотите поместить резервуар. Внешний вид резервуара показан на Рис.2.23.



Рис.2.23

Если щелкнуть левой кнопкой мыши по блоку, то он будет выделен синим цветом. В этом состоянии можно изменить название резервуара. Если вы хотите, чтобы название располагалось снизу или сбоку от резервуара, то надо мышью щелкнуть по названию, нажать на клавишу **Ctrl** и, удерживая их нажатыми, перенести название в нужное место. Двойной щелчок на резервуаре открывает диалоговое окно определения резервуара (Рис.2.24).



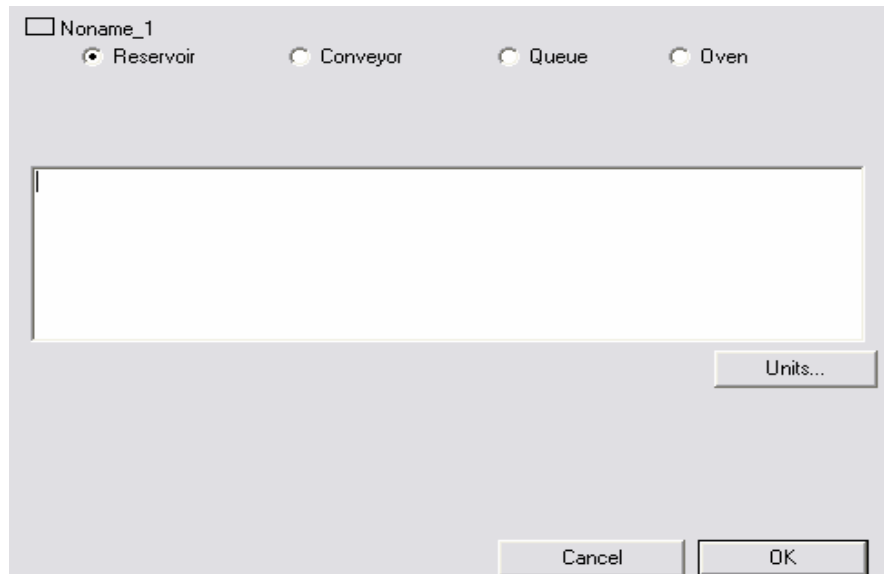



Рис.2.24

В этом окне можно выбрать тип фонда с помощью соответствующего переключателя. В поле можно записать заметки и комментарии. С помощью кнопки **Units** задаются единицы измерения.

Такое диалоговое окно появляется, если вы находитесь на этапе составления схемы модели. После нажатия на кнопку , когда на иконке резервуара появляется изображение знака вопроса (?), после двойного щелчка мышью появляется диалоговое окно определения числовых параметров резервуара (Рис.2.25).

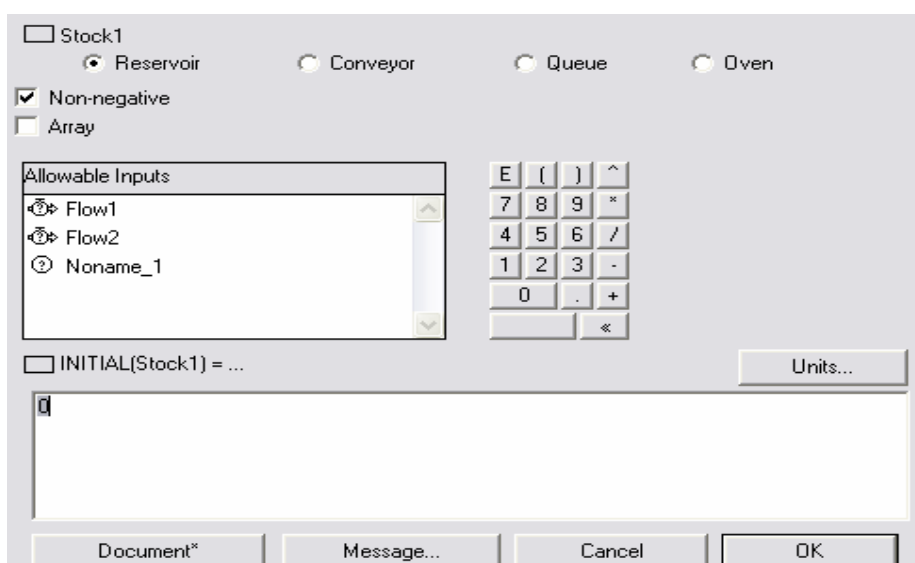


Рис.2.25

Если флажок **Non-negative** включен, то значение резервуара не может быть отрицательным. Флажок **Array** используется для повторения аналогичных деталей при наличии одинаковых цепей в модели. В списке **Allowable Inputs** представлены доступные входные переменные для определения начального значения резервуара. Начальное значение может быть константой или алгебраическим выражением. Вводить числа и знаки операций можно как с клавиатуры, так и с использованием кнопок в диалоговом окне. С помощью кнопки **Units** задаются единицы измерения. Кнопка **Document** позволяет получить доступ к комментариям и заметкам. Звездочка (\*) означает, что заметки присутствуют в блоке. Кнопка **Message** позволяет послать сообщения.

### 2.3.2.3. Конвейер

**Конвейер** получает материал из потока, хранит его в течение определенного времени (то есть пока он находится на ленте конвейера), затем выгружает. Время работы конвейера может задаваться как константой, так и переменной. Входящий в конвейер поток должен быть однонаправленным.

Создать конвейер можно двумя способами:

- При нажатии на кнопку **резервуар** из открывающейся палитры выбрать элемент конвейер, а затем щелкнуть в том месте рабочего листа, в котором вы хотите поместить этот элемент.

- Создать резервуар, а затем в диалоговом окне определения резервуара выбрать переключатель **Conveyor**.

Внешний вид конвейера показан на Рис.2.26. Диалоговое окно определения конвейера аналогично окну определения резервуара.

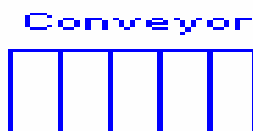


Рис.2.26

Вид диалогового окна определения числовых параметров конвейера зависит от входящих в него потоков. Существует три возможных ситуации.

1. Входной поток выходит из резервуара и/или облака неограниченности. Внешний вид диалогового окна показан на Рис.2.27.

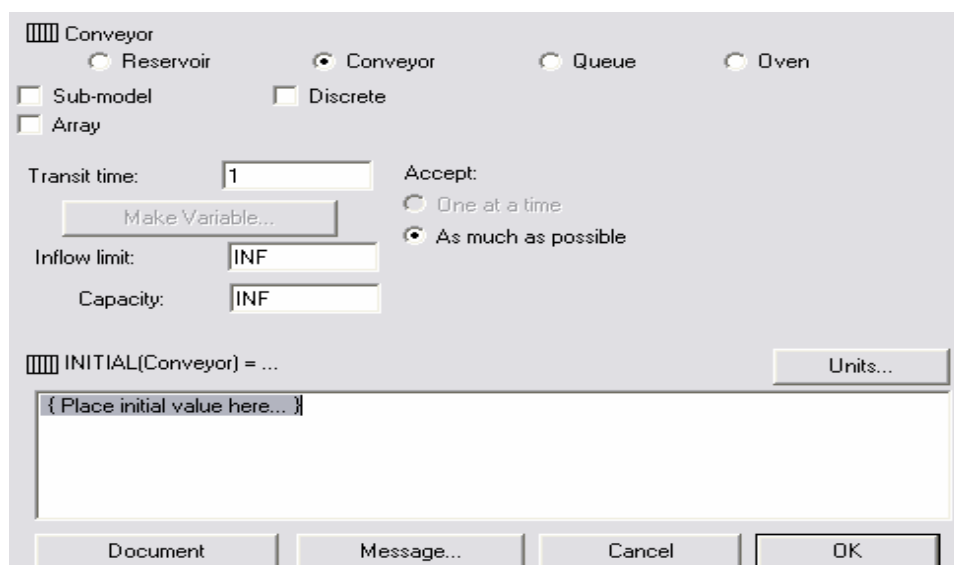


Рис.2.27

Флажок **Sub-model** позволяет превратить конвейер в подмодель. Флажок **Array** используется для повторения аналогичных деталей при наличии одинаковых цепей в модели. Когда отмечен флажок **Discrete**, конвейер имеет дело с дискретными величинами. В поле **Transit time** вводится постоянная, которая описывает время, в течение которого объект находится внутри конвейера. Переключатель **As much as possible** означает, что конвейер может принять так много материала, как возможно. Это количество определяется логикой потока, ограниченного предельным значением входного потока и емкостью конвейера. Данные параметры задаются в полях **Inflow limit** и **Capacity** соответственно. По умолчанию значение этих величин равно бесконечности. В поле инициализации конвейера задается начальное значение. Существует два метода инициализации конвейера:

- Конвейер инициализируется одним числом. Оно равномерно распределяется на всю длину конвейера.

- Серия чисел разделенных запятой, по одному для каждой единицы во времени транзита.

2. Входной поток выходит из очереди, но при условии, что нет входных потоков из конвейеров и/или печей.

Диалоговое окно определения числовых параметров конвейера показано на Рис.2.28:

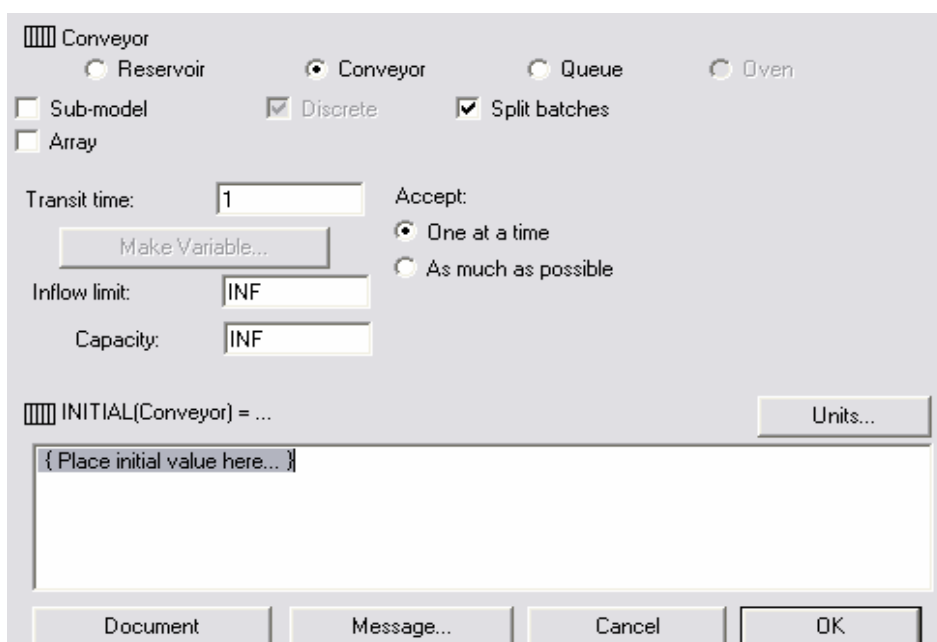


Рис.2.28

В этом случае конвейер может иметь дело, только с дискретными величинами. Флажок **Split batches** позволяет конвейеру взять из очереди следующие значения, каждый раз, когда он превышает предел входного потока или емкость. Если флажок выключен, то поток, выходящий из очереди, будет установлен на 0, если значение потока превышает установленные пределы. Если включен переключатель **One at a time**, то конвейер будет брать один элемент из очереди в каждый временной период.

3. Входной поток выходит из конвейера или печи.

На Рис.2.29 представлено диалоговое окно определения числовых параметров конвейера. Заполняется аналогично всему вышесказанному. Поля **Inflow limit** и **Capacity** являются неактивными, потому что входящий поток полностью определяется вышестоящим конвейером или печью. Кнопка **Why Grayed** объясняет этот факт. Также поэтому активен только переключатель **As much as possible**.

Из конвейера может выходить не больше двух потоков. При этом первый является прямым потоком, а второй – потоком утечки.

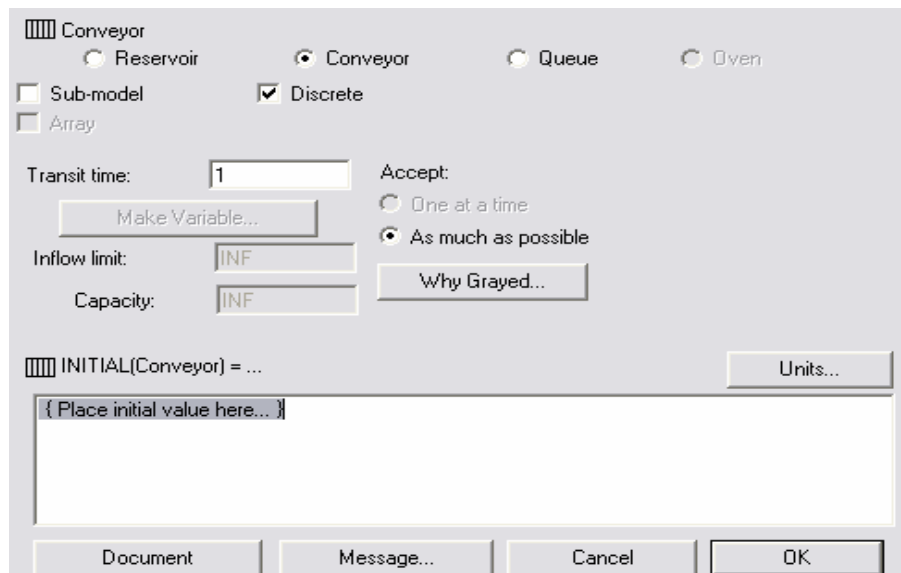


Рис.2.29

Основной вид диалогового окна для определения числовых параметров выходного потока представлен на Рис. 2.30:

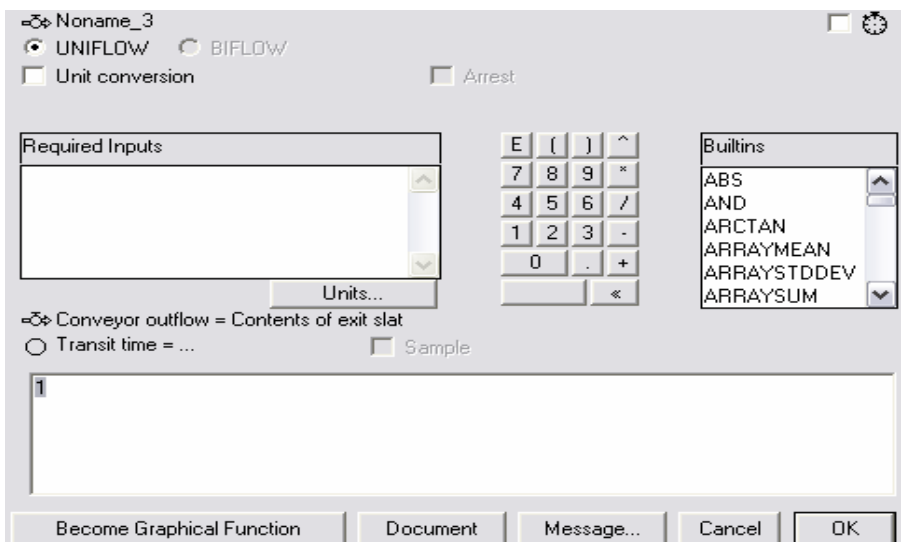


Рис.2.30

Из Рис.2.30 видно, что это окно имеет такой же вид, как и окно обычного потока. Различие состоит в том, что в поле инициализации указывается значение времени транзита. Оно может быть как постоянным, так и переменным. Задать переменное значение можно двумя способами:

ü В диалоговом окне определения параметров конвейера нажать на кнопку **Make Variable**.

ü Ввести функцию в поле инициализации времени транзита.

Если в списке **Required Inputs** есть элементы, то становятся активными флажки **Arrest** и **Sample**. Опция **Arrest** позволяет остановить операции на конвейере, если значение выбранной из списка переменной равно 1. После того, как вы поставите флажок, то появится поле, в котором и надо указать название переменной из **Required Inputs** (Рис.2.31):

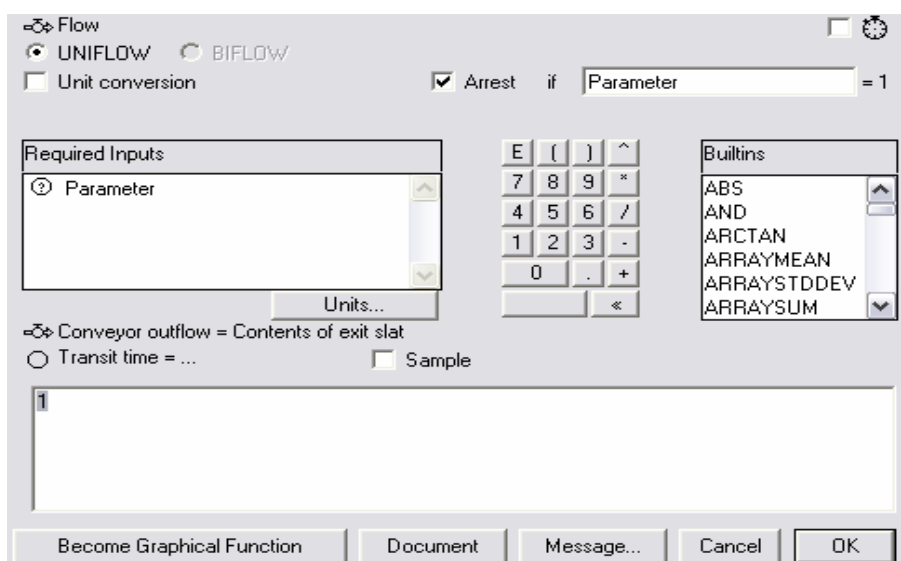


Рис.2.31

Когда поставлен флажок **Sample**, значение времени транзита по формуле из поля инициализации подсчитывается каждый раз, когда конвертор **Parameter** генерирует значение 1 (Рис.2.32).

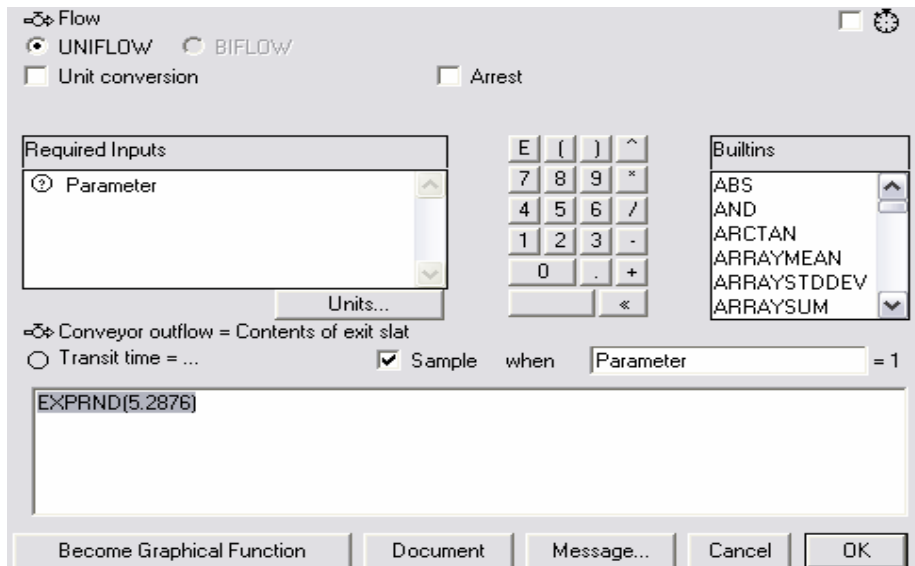


Рис.2.32

В модели может быть утечка содержания из конвейера, которая происходит через второй выходной поток. Диалоговое окно определения потока утечки представлено на Рис.2.33:

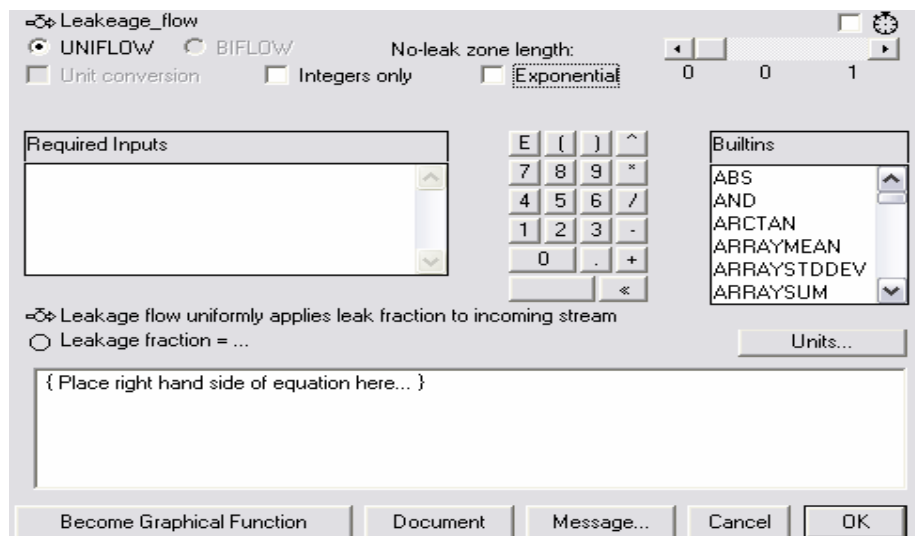


Рис.2.33

С помощью слайдера выбирается та доля времени транзита, которая не доступна для утечки. Если установлено значение 0, то это значит, что конвейер склонен к утечке; если значение 1, то утечка происходит только после того, как конвейер наполнится целиком.

Флажок **Exponential** создает экспоненциальное распределение утечки вместо равномерного распределения. Флажок **Integers only** округляет значение потока утечки до целого числа в каждый момент времени. Флажок **Unit conversion** является активным, если поток соединяет конвейер с другим фондом.

В поле инициализации задается доля утечки. Она может быть определена с помощью константы, списка **Required Inputs**, алгебраических функций и операций, графической функции. Доля утечки должна лежать в интервале от 0 до 1.

#### 2.3.2.4. Очередь

В процессе моделирования особая роль отводится блоку **очередь**. Он способен сформировать последовательность дискретных сигналов, олицетворяющих либо отдельные объекты, либо “порции” какого-либо материала. Порядок входа и выхода определяется принципом **FIFO** – объект, первым поступивший на склад, первым его покидает. Это соответствует наиболее распространенному бухгалтерскому принципу учета товарно-материальных запасов. Однако важнейшая особенность очереди в том, что она способна направлять избыточное количество материала “свободным” или недогруженным объектам в соответствии с установленной очередностью.

Создать очередь можно двумя способами (также как и конвейер):

- При нажатии на кнопку **резервуар** из открывающейся палитры выбрать элемент очередь, а затем щелкнуть в том месте рабочего листа, в котором вы хотите поместить этот элемент.

- Создать резервуар, а затем в диалоговом окне определения резервуара выбрать переключатель **Queue**.

Внешний вид конвейера показан на Рис.2.34. Диалоговое окно определения очереди аналогично окну определения резервуара.

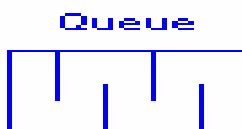


Рис.2.34

Диалоговое окно для определения числовых параметров очереди заполняется очень просто (Рис.2.35):



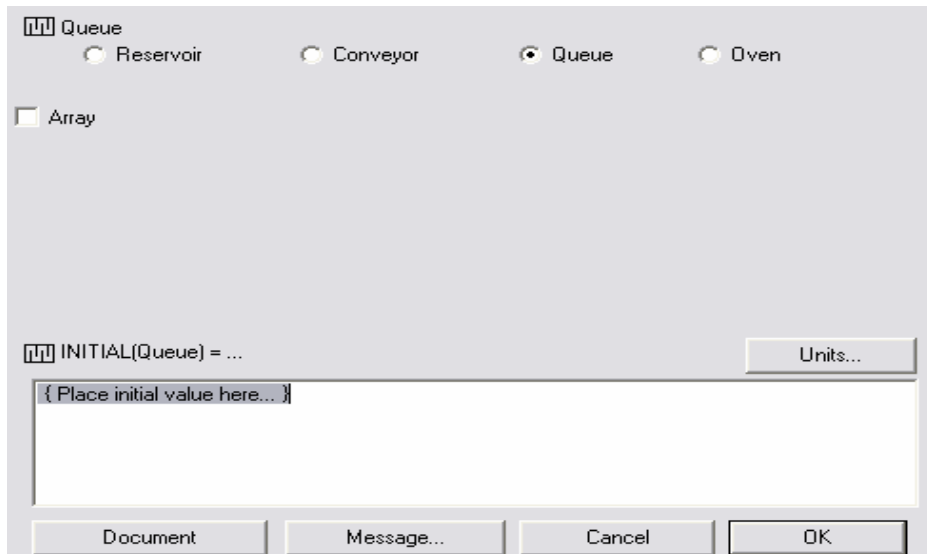


Рис.2.35

Инициализировать очередь можно двумя способами:

- ü Задать 0, если очередь в начальный момент времени пуста.
- ü Можно ввести последовательность положительных чисел, разделенных запятой. Надо отметить, что первое число, которое вы ввели, будет первым и на выходе из очереди.

Выходной поток очереди должен быть однонаправлен. Заметим, что этот поток будет ограничен только в том случае, если вниз по течению модели находится конвейер или печь с ограничениями на входной поток или емкость. Конверсия потока возможна в том случае, если он соединяет очередь с другим фондом. Диалоговое окно определения числовых параметров потока приведено на Рис.2.36:

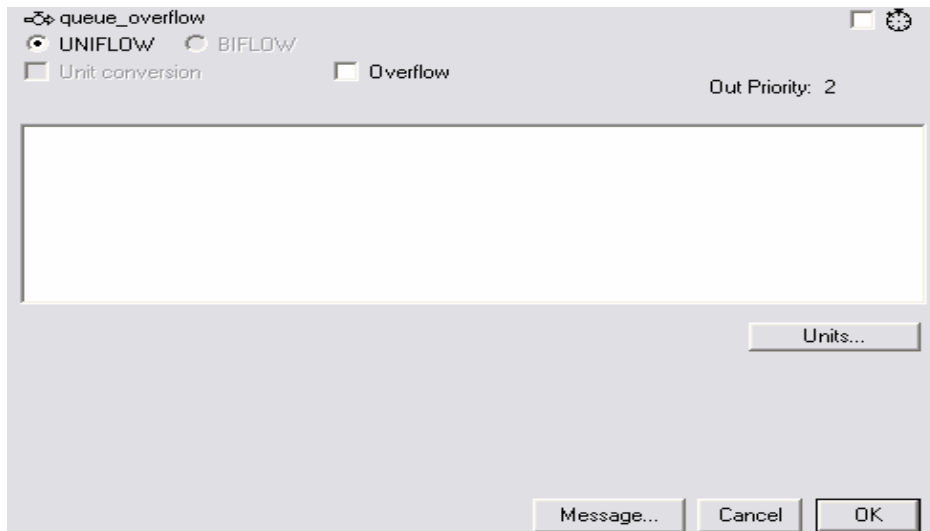


Рис.2.36

Флажок **Overflow** появляется, если поток выходит из очереди. Если флажок отмечен, то всякий раз, когда поток с более высоким приоритетом блокирован (то есть он наполнен при условии наличия ограничений), выход из очереди направляется на поток с более низким приоритетом. В нашем случае приоритет потока равен 2.

### 2.3.2.5. Печь

Один из наиболее сложных типовых структурных строительных блоков - **печь**. Он предусматривает различные режимы и сроки “переработки” поступивших в него объектов или порций материала. Элемент печь предназначен для обработки дискретных данных.

Создается печь аналогично элементам конвейер и очередь. Внешний вид блока печь показан на Рис.2.37:



Рис.2.37

Входной поток печи должен быть однонаправленным. Диалоговое окно для определения числовых параметров печи с входным потоком из облака неограниченности или резервуара приведено на Рис.2.38.

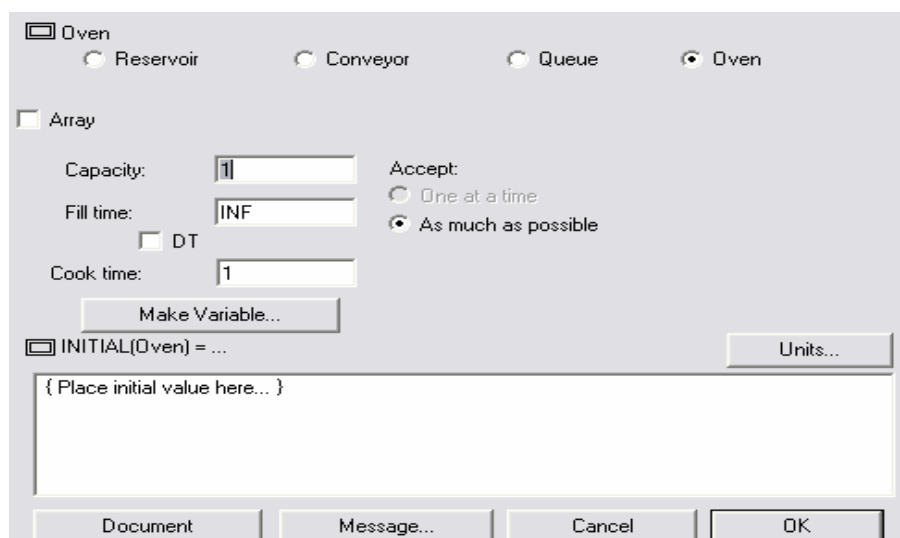


Рис.2.38

В поле **Capacity** указывается емкость печи, то есть количество материала, которую может содержать печь. Печь закрывает доступ и начинает «готовить» содержимое, если заполнена вся емкость или истекло доступное время (учитывается то, что происходит первым). В поле **Fill time** указывается доступное время, то есть то время, когда печь принимает материал. Если отметить флажок **DT**, то значение доступного времени будет равно единице времени дискретной модели. Переключатель **As much as possible** означает, что конвейер может принять так много материала, как возможно. Это количество определяется логикой потока, каждый раз когда печь принимает материал.

В поле **Cook time** указывается время, в течение которого материал находится в печи. Оно может быть как постоянным, так и переменным, аналогично времени транзита для конвейера. Печь инициализируется неотрицательным числом, которое не может быть больше ее емкости.

Переключатель **One at a Time** становится активным, если входной поток выходит из очереди. Он означает, что печь берет следующий пункт из очереди в каждый момент времени, согласно ограничениям емкости. Также в этом случае в диалоговом окне появляется флажок **Split batches**, который позволяет печи взять из очереди следующие значения, каждый раз, когда он превышает емкость.


На Рис.2.39 представлен основной вид диалогового окна определения числовых параметров для потока, выходящего из очереди.



Рис.2.39

Заполнение этого окна аналогично заполнению диалогового окна определения числовых параметров для потока, выходящего из конвейера. Для элемента печь процесс может быть остановлен с использованием флажка **Arrest**, также как для элемента конвейер. Этот флажок активен, если список **Required Inputs** не пуст.

### 2.3.2.6. Поток

**Поток** - это процесс, протекающий непрерывно во времени, оценить который можно в физических или денежных единицах, соотнесенных с каким-либо временным интервалом (рубли в месяц, литры в час, стоимость акций на время закрытия биржи в данный день и т.д.). В **Ithink** поток изображается фигурой, состоящей из путепровода, вентиля, регулятора потока и указателя направления. Чтобы создать поток, нажмите на кнопку , затем нажмите на левую кнопку мыши в том месте, откуда выходит ваш поток, и проведите до того места, где он заканчивается. Это может быть фонд, если поток ограничен, или любая точка рабочего листа, если поток неограничен. Если вы хотите изогнуть поток, то во время рисования нажмите клавишу **Shift**.

По характеру использования, потоки подразделяются на ограниченные и неограниченные, однонаправленные и двунаправленные, конвертируемые и неконвертируемые.

Поток, как правило, ограничивается фондом. Однако нередки случаи, когда модельная ситуация требует использования неограниченных потоков, и тогда **Ithink** прикрепляет к соответствующему источнику или приемнику «облако» неограниченности. Действительно, как можно ограничить поток заказов при стихийно складывающихся

ценах на рынке потребительских товаров? Какими рамками можно ограничить степень доверия к той или иной финансовой компании? Наконец, каким образом количественно выразить границы существования таких идеализируемых концептов, как поток возможных инвестиций, predisposedность представителей группы риска к потреблению наркотиков, перманентное и неконтролируемое истощение природных ресурсов и т.д. Внешний вид потоков показан на Рис.2.40.

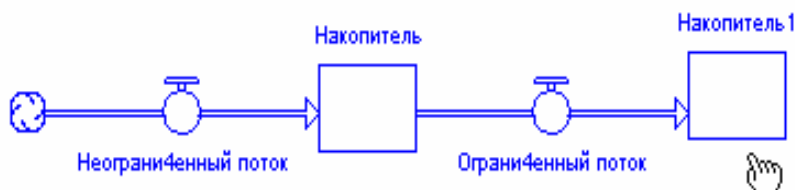


Рис.2.40

Кроме того, встречаются модельные ситуации, требующие использования двунаправленных потоков (Рис.2.41). Например, цены на рынке колеблются, а Вам желательно использовать лишь один поток, характеризующий эти колебания. Или, например, Вам хотелось бы исследовать динамику изменения заработной платы, которая не только будет расти в связи с инфляцией, но и падать в зависимости от конъюнктуры рынка труда и складывающихся цен на потребительские товары и услуги. Двунаправленный поток можно создать двумя способами: либо включить соответствующий переключатель в окне определения потока, либо щелкнуть на маленькой стрелке в правом нижнем углу кнопки с изображением потока и из открывшейся палитры выбрать соответствующий элемент.

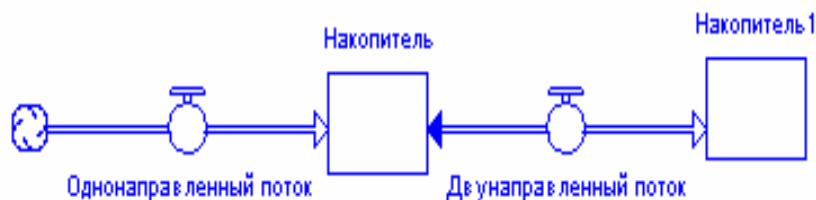


Рис.2.41

Наконец, потоки иногда требуют конвертирования своего содержимого для того, чтобы поставить в соответствие объемы соединяемых потоками фондов (Рис.2.42).

Действительно, они соотносятся, как правило, в отношении 1:1 (наличные деньги налогоплательщиков и государственный бюджет, материальные запасы оптовика и наличие товаров у розничных торговцев, объем хранилища топлива на бензозаправочной станции и объем бензобака при заправке и т.д.). Но без конвертирования трудно соотнести количество комплектующих к числу собранных агрегатов при монтаже; или количество бревен к числу досок при распилке (соотношение 1:n).

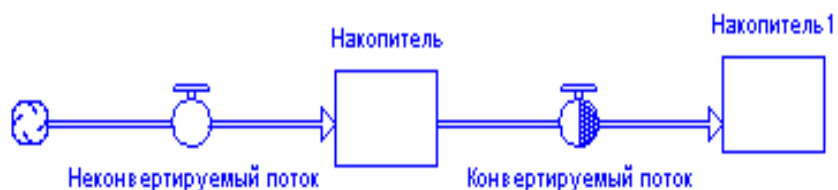


Рис.2.42

После двойного щелчка на изображении потока открывается диалоговое окно определения потока (Рис.2.43).

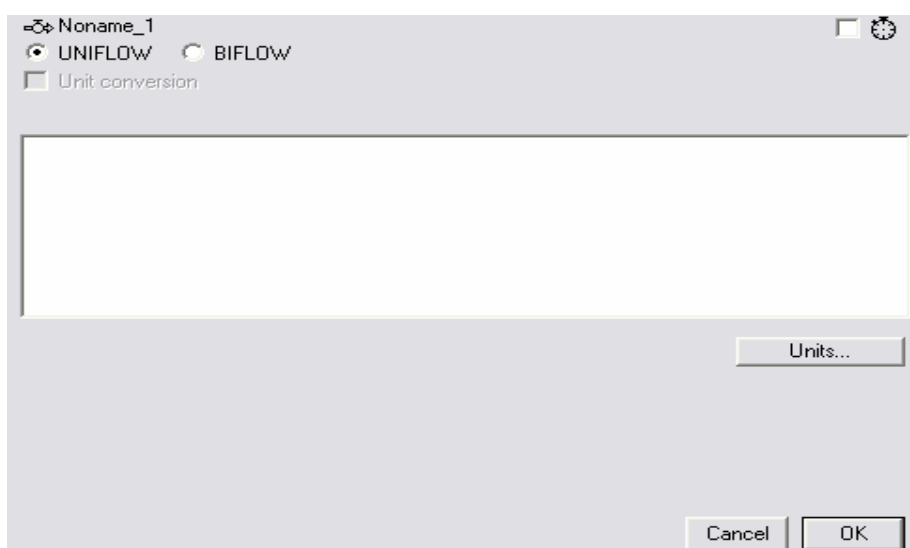


Рис.2.43

Это окно заполняется аналогично окну определения резервуара. Флажок **Unit conversion** можно использовать для задания конвертирования только в том случае, если поток соединяет два фонда.

Диалоговое окно определения численных параметров потока представлено на Рис.2.44.

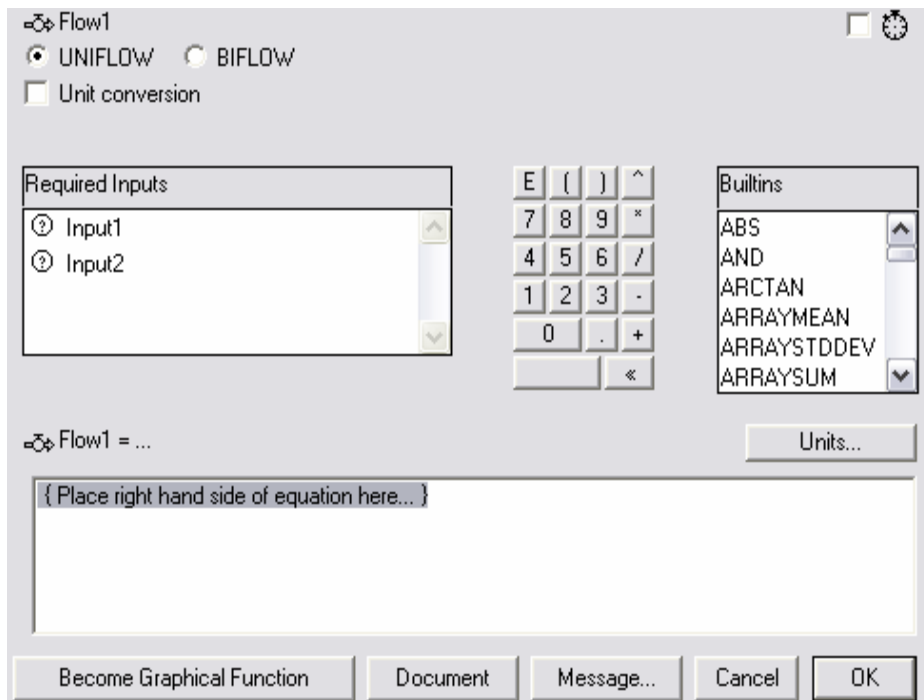


Рис.2.44

С помощью калькулятора можно вводить числа и символы. В списке **Required Inputs** перечислены параметры, которые могут использоваться для определения потока (это конверторы, которые соединены с данным потоком коннекторами). Если вы щелкаете по названию переменной, то оно появляется в окне инициализации потока. В списке **Builtins** перечислены все доступные функции, которыми вы можете пользоваться для определения потока. Надо отметить, что в пакете ITHINK достаточно много встроенных функций. Кнопки **Document**, **Message** и **Units** были описаны выше. С помощью кнопки **Become Graphical Function** поток можно определить как графическую функцию. (Подробнее об этом будет сказано в разделе Конвертор). В случае если вы хотите конвертировать значение потока и отметите флажок **Unit conversion**, то диалоговое окно немного изменится: появится новое поле, в котором указывается значение множителя. Это может быть как число, так и некоторый параметр из списка **Required Inputs** (Рис.2.45).

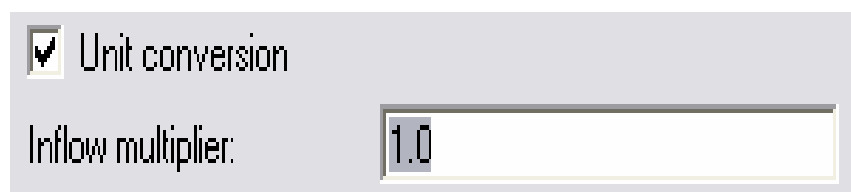


Рис.2.45

### 2.3.2.7. Конвертер

**Конвертеры** в **Ithink** – преобразователи модельных единиц, которые Вы можете использовать для детализации и уточнения поведения потоковых схем. Изображаются они окружностями (Рис.2.46).

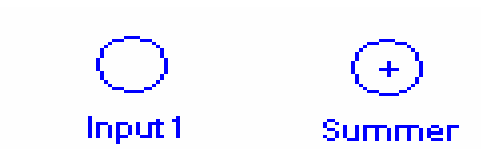



Рис.2.46

Они могут содержать значения констант или внешних входных переменных, подсчитывать значения алгебраических выражений или использоваться для хранения графических функций. Для создания конвертера надо нажать кнопку , а потом щелкнуть по нужному месту нашего рабочего листа. Отметим, что в открывающейся палитре присутствует специальный вид конвертера – сумматор. Он позволяет суммировать любые параметры модели, не требуя рисования коннекторов. Диалоговое окно определения конвертера содержит только поле ввода заметок и кнопку **Units** для определения единиц измерения, поэтому заполняется так, как было сказано выше. Диалоговое окно определения числовых параметров конвертера показано на Рис.2.47.

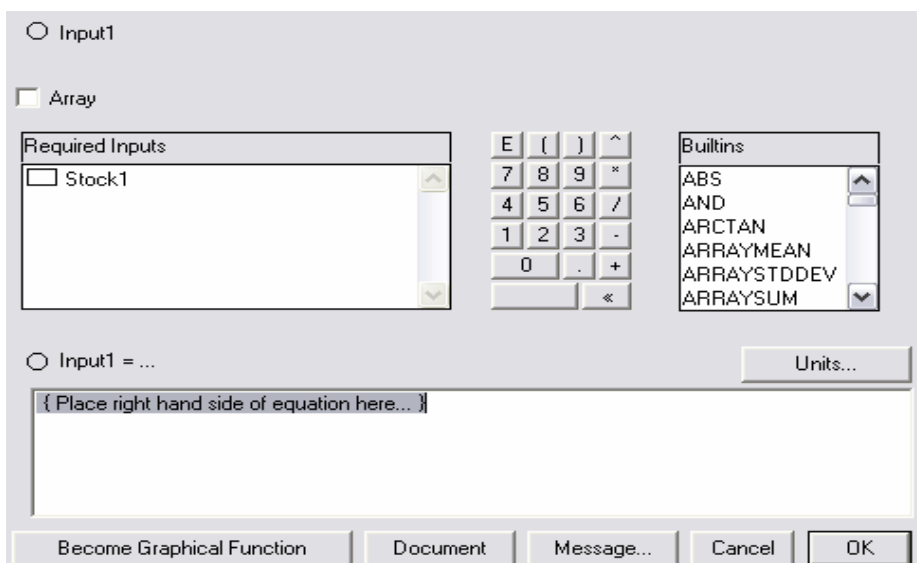


Рис.2.47



Это окно заполняется аналогично окну определения числовых параметров потока. Рассмотрим кнопку **Become Graphical Function**, с помощью которой можно задать графическую зависимость.

**Графическая функция** – это эскиз взаимосвязи между некоторым входным параметром (из списка **Required Inputs** или **Builtins**) и выходом. Выбираем из указанных списков входной параметр (например, функцию **TIME**, которая является временем в модельной имитации) и щелкаем по нему, он появляется в поле инициализации конвертора. Затем нажимаем на кнопку **Become Graphical Function**. Появляется диалоговое окно определения графической функции (Рис.2.48).

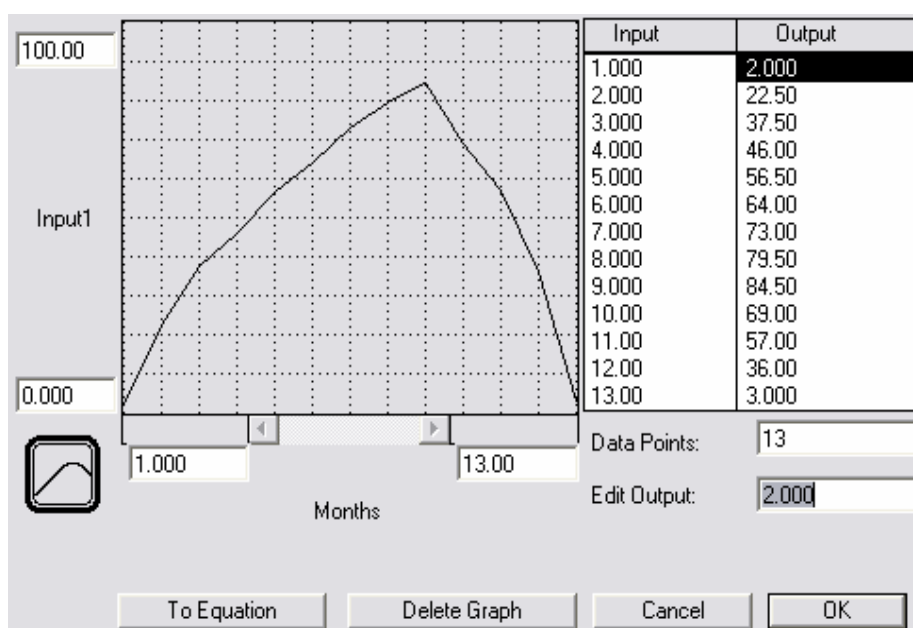


Рис.2.48

В полях, которые расположены рядом с осями, вводится минимум и максимум значений параметров на осях. График можно вводить двумя способами:

- Щелкнуть мышкой внутри сетки, при этом появляются линии, которые соединяют заданные точки.

- Задавать значения координат по оси Y в поле **Edit Output**. Для этого надо сначала выделить в таблице то значение входа, для которого вы хотите ввести выходное значение, а потом в поле **Edit Output** ввести это значение.

С помощью кнопки **Delete Graph** график можно удалить, а с помощью кнопки **To Equation** вернуться в окно определения числовых параметров конвертера. В поле **Data Points** вводится количество точек, для которых вы будете задавать координаты **Y**. Если количество точек больше 13, то график будет иметь полосу прокрутки. Всего может быть до 1500 точек. Чтобы увидеть график целиком, нажмите клавишу **Alt**. Кнопка, которая находится около начала координат, определяет вид графика: непрерывный или разрывной (Рис.2.49).

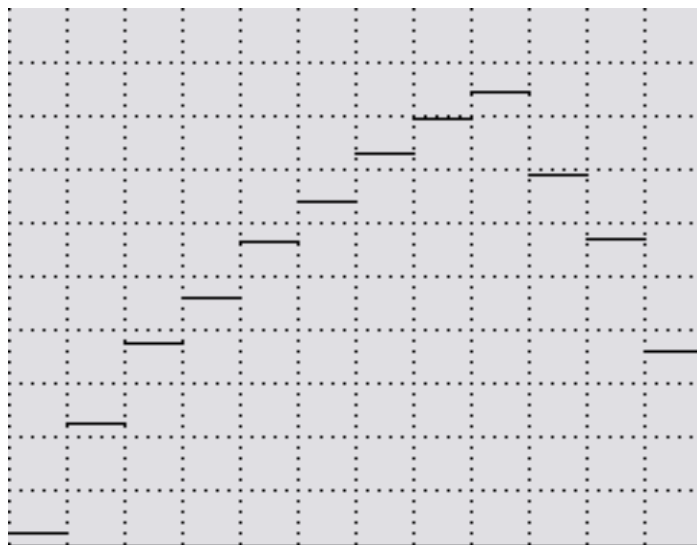


Рис.2.49

Диалоговое окно определения числовых параметров сумматора (Рис.2.50) немного отличается от диалогового окна определения конвертера. Оно доступно только в том случае, если у конвертера нет входных параметров.

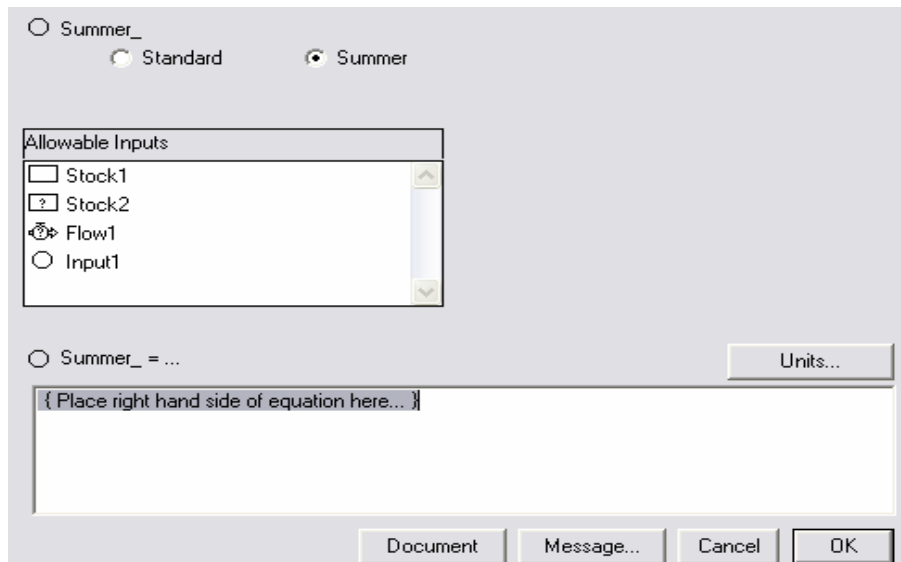



Рис.2.50

С помощью переключателя задается тип конвертера: стандартный или сумматор. Из списка **Allowable Inputs** выбираются те блоки модели, которые мы хотим суммировать.

### 2.3.2.8. Коннектор

**Коннектор** предназначен для связи между собой элементов модели. Существуют два вида коннекторов: действующий коннектор (помечается сплошной розовой стрелкой) и информационный коннектор (помечается пунктирной розовой стрелкой) (Рис.2.51). Надо отметить, что коннектор не может указывать на фонд, так как его значение можно изменять только с помощью потока.

Чтобы нарисовать коннектор, надо нажать на кнопку . Вид коннектора можно выбрать из открывающейся палитры. Рисуются коннектор также как поток, но он обязательно должен связывать между собой два строительных блока.

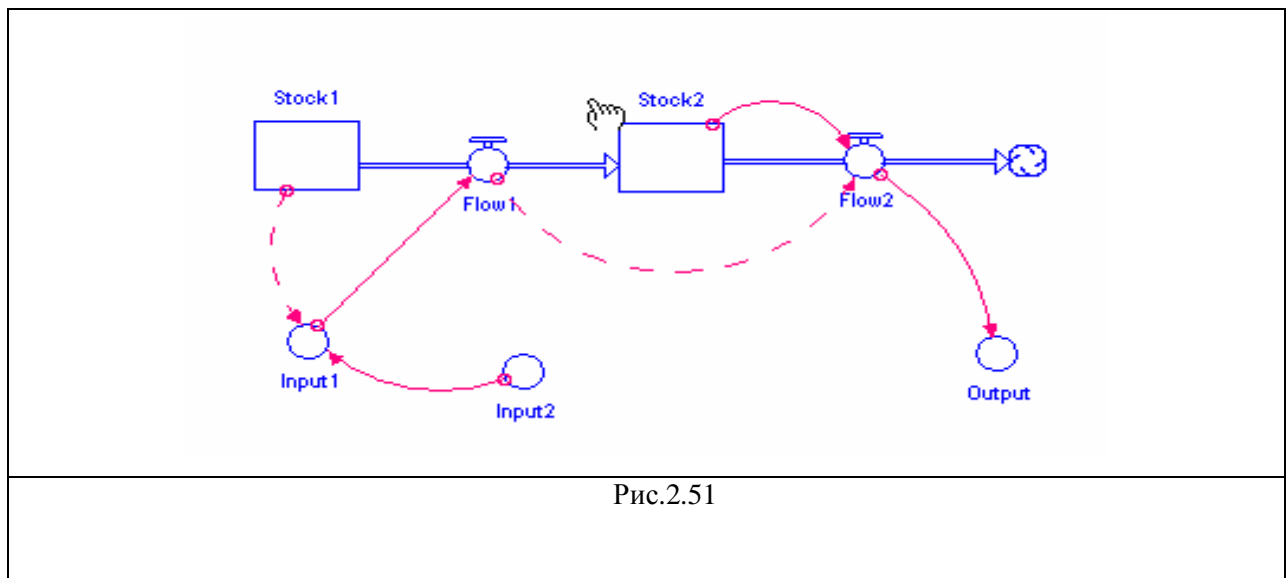


Рис.2.51

На Рис.2.51 показаны некоторые виды коннекторов. Следует отметить, что при создании коннекторов в модели у вас могут возникнуть окна с предупреждениями:

ü Если вы пытаетесь зациклить модель, то есть соединить первый блок цепи с последним, возникнет окно, представленное на Рис.2.52. Фатальное зацикливание запрещено в пакете **Ithink**, так как может привести к бесконечной работе модели.

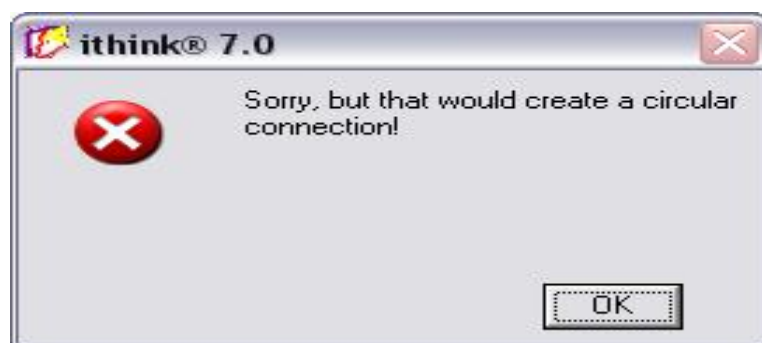


Рис.2.52

ü Если вы уже нарисовали один коннектор, и пытаетесь соединить те же блоки вторым, то возникнет окно (Рис.2.53):

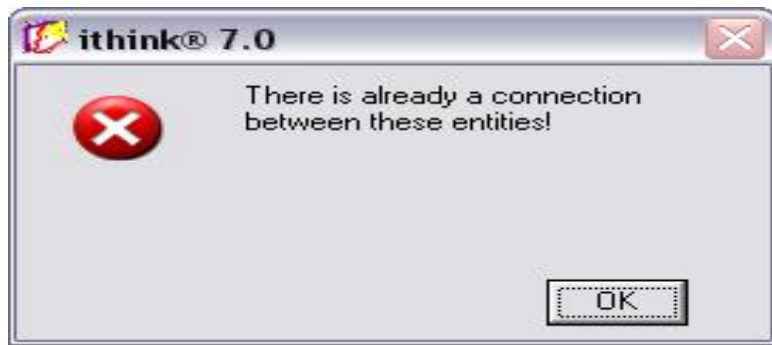


Рис.2.53

### 2.3.2.9. Ромбоидальное определение процесса

**Ромбоидальное определение процесса (РОП)** – это механизм для управления запутанными схемами, связанный с представлением определения процессов внутри модели. С помощью этого блока можно скрыть сложность определенных правил, направляя потоки в своеобразный «черный ящик» (на самом деле, бледно-лиловый ромб, Рис.2.54). Внешне можно увидеть входные и выходные потоки, связанные с этим блоком. При желании можно просмотреть и те связи, которые находятся внутри блока.

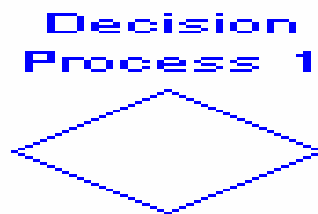



Рис.2.54

Чтобы создать РОП, надо нажать кнопку  на панели инструментов, а потом щелкнуть в том месте рабочего листа, где вы хотите поместить этот элемент. Когда вы откроете РОП, то есть два раза щелкните по нему мышкой, вы получите некоторое пространство, внутри которого можно поместить определенную логическую схему, управляющую потоком (Рис.2.55).

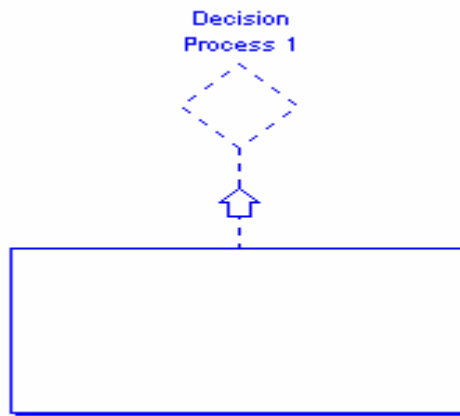


Рис.2.55

Если щелкнуть по стрелке вверх левой кнопкой мыши, то рамка будет скрыта внутри ромба. Надо отметить, что когда рамка открыта, то все блоки модели, которые находятся за пределами рамки, становятся пунктирными, а рука приобретает вид перечеркнутого круга (⊘). Чтобы вернуть ей нормальный вид для выделения каких-либо элементов вне рамки, надо нажать клавишу **Ctrl**.

После того, как вы поместили ромбоидальное определение процесса в модели, первым шагом будет использование действующие и информационные коннекторы для указания всех основных связей, входящих и выходящих из определения процесса. Надо отметить, что вход в РОП – это всегда информационный коннектор.

После того, как вы создали основную схему модели и провели действительные и информационные коннекторы (Рис.2.56), вы можете переходить к созданию логических схем на микроуровне внутри РОП.

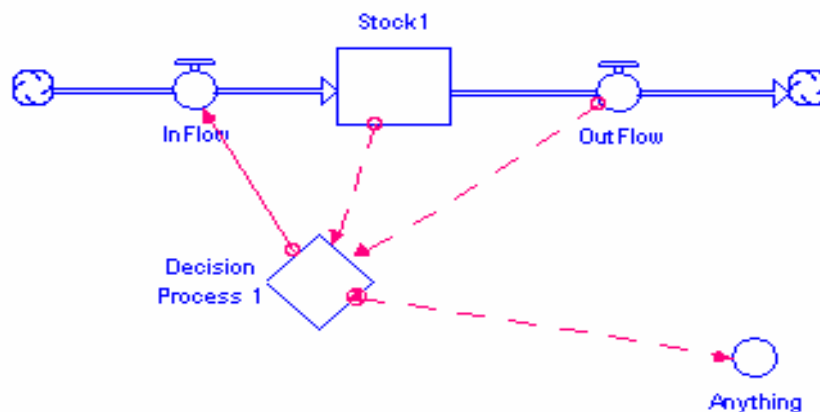


Рис.2.56

Как видно из Рис.2.57, внутри рамки находятся розовые кружочки, которые отражают соединения, которые вы сделали при составлении модели. Чтобы связать их с процессом, просто перенесите кружочки на соответствующие элементы пространства РОП.

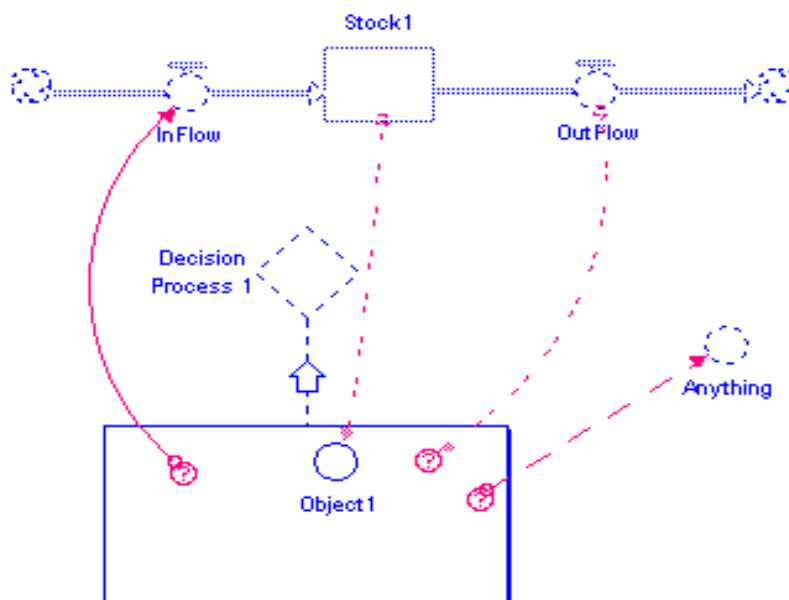


Рис.2.57

Схема модели внутри пространства РОП рисуется так же, как и обычная схема. Но следует отметить следующие факты:


- Схемы должна быть замкнутой. Соединить с внешней моделью можно только тем способом, который указан выше.
- Внутри рамки нельзя разместить рамку сектора или кнопку.
- Нельзя создать подмодель или ромбоидальное определение процесса внутри пространства РОП.

## 2.4. Инструменты


Далее описываются основные инструменты пакета **Ithink**, которые позволяют осуществлять основные действия над объектами модели: выбор, перемещение, окраска, удаление и определение значений.

Всего существует четыре инструмента: *рука*, *кисть динамит*, *фантом*.


### 2.4.1. Инструмент «Рука»

Инструмент **Рука** – многоцелевой инструмент редактирования (кнопка ). Он используется для выбора, перемещения, доступа и редактирования строительных блоков и объектов. Также его используют для управления подмоделями и выбора пунктов на уровне программного кода. Для осуществления этих действий надо направить указательный палец **Руки** на необходимый объект и щелкнуть левой кнопкой мыши. Данный инструмент является выбранным по умолчанию. После выполнения какого-либо действия с помощью другого инструмента приложение передает управление инструменту **Рука**.


#### 2.4.2. Инструмент «Кисть»

Инструмент **Кисть** (кнопка ) позволяет окрасить элементы модели: строительные блоки, объекты и фон. Подержав некоторое время мышью указатель на кончике кисти пиктограммы, можно вызвать палитру цветов. Далее мышью переведите кисть на строительный блок, который нуждается в раскраске, и один раз щелкните мышью. Блок окрасится выбранным цветом.

#### 2.4.3. Инструмент «Динамит»

Инструмент **Динамит** (кнопка ) используется для удаления блоков и объектов.

#### 2.4.4. Инструмент «Фантом»

Инструмент **Фантом** (кнопка ) доступен только на уровне модели. Он создает копию строительного блока (фонда, потока или конвертера), которая не имеет собственного уравнения. **Фантом** является чрезвычайно эффективным инструментом при конструировании топологически сложных потоковых схем. В таких схемах коннекторы настолько перепутываются, что невозможно отследить и наглядно представить взаимодействия. Копии объектов позволяют "распутывать" и упрощать исходные потоковые схемы, делая их наглядными и структурированными.

### 2.5. Объекты


В пакете **Ithink** существуют различные объекты, которые позволяют достичь наглядности в нашей работе и вывести результаты работы модели. Семь из них доступны только на уровне интерфейса. Это **Цикл**, **Индикатор состояния**, **Переключатель**, **Устройство графического ввода данных**, **Ручка**, **Устройство для ввода данных в виде таблицы** и **Слайдер**. **Фрейм сектора** доступен только на модельном уровне. Остальные шесть объектов: **Кнопка**, **График**, **Таблица**, **Текст**, **Рисунок** и **Числовой дисплей** доступны и на модельном уровне, и на уровне интерфейса. Ниже будут рассмотрены наиболее часто используемые объекты.



### 2.5.1. Фрейм сектора

**Фрейм сектора** используется для группирования элементов модели. Например, если вы строите модель предприятия, то ее можно разделить на производственный сектор, управленческий сектор, финансовый сектор и сектор рабочей силы. Создать **фрейм сектора** можно двумя путями:

• Если отмечен флажок **Link High-Level Map to Model** в пункте меню **Interface Prefs** на уровне интерфейса, то при создании фрейма на уровне интерфейса **сектор фрейма** на модельном уровне будет создан автоматически.

• На модельном уровне нажать на кнопку , а потом щелкнуть в том месте рабочего листа, где вы хотите поместить этот элемент.

Надо отметить, что программа не позволит вам наложить один сектор на другой.

Внешний вид **фрейма сектора** показан на Рис. 2.58.

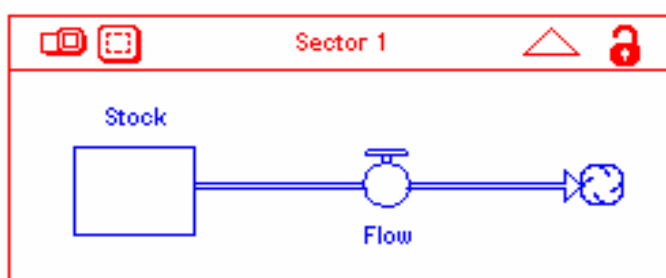


Рис.2.58

Крайняя левая кнопка в заголовке сектора делает **фрейм сектора** прозрачным/непрозрачным. Вторая кнопка слева позволяет выделить все элементы, которые находятся внутри сектора. Стрелка вверху справа обеспечивает переход к соответствующему фрейму на уровень интерфейса. В правом верхнем углу находится замок. Когда фрейм закрыт, то программа не позволит уменьшить фрейм до такой степени, что какие-либо элементы выдут за пределы фрейма. Кроме того, если вы растягиваете запертый фрейм сектора, и в него попадают какие-либо, ранее не принадлежавшие ему, элементы, то они не войдут в сектор и будут заштрихованы.

Чтобы открыть диалоговое окно определения **фрейма сектора** необходимо два раза щелкнуть мышкой по рамке или по заголовку фрейма (Рис.2.59).

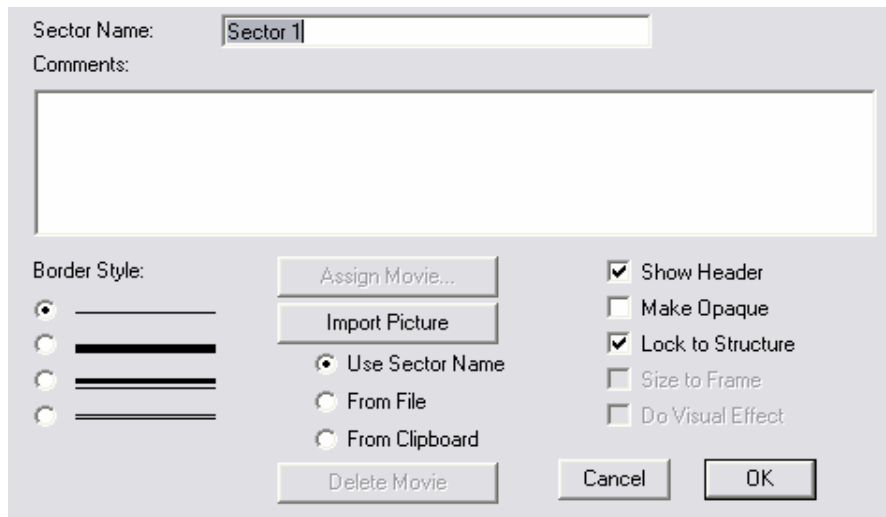


Рис.2.59

В этом окне можно определить имя сектора, задать некоторые пояснения, выбрать стиль рамки, вставить картинку (это выполняется аналогично вставке картинки во фрейм на уровне интерфейса). Если флажок **Show Header** не будет отмечен, то заголовок сектора будет скрыт. Флажки **Make Opaque** и **Lock to Structure** дублируют кнопки, которые находятся в заголовке сектора и позволяют сделать сектор непрозрачным и закрытым соответственно. Флажок **Size to Frame** является активным, если во фрейм сектора вставлена картинка. Он позволяет полностью поместить картинку внутри фрейма. Флажок **Do Visual Effect** позволяет визуализировать процесс перехода сектора от прозрачного состояния к непрозрачному.

### 2.5.2. Текст


Чтобы вставить в схему модели какие-либо пояснения, используется объект **текст**. Для того чтобы создать текст, надо нажать кнопку  на панели инструментов, а потом щелкнуть в нужном месте рабочего листа. Левый верхний угол текстовой рамки будет размещен в том месте, где вы щелкнули мышкой (Рис.2.60).



Рис.2.60

Если щелкнуть мышкой внутри рамки, то появится курсор и можно начинать вводить текст. Чтобы открыть диалоговое окно определения текста, надо два раза щелкнуть на рамке текста (Рис.2.61).

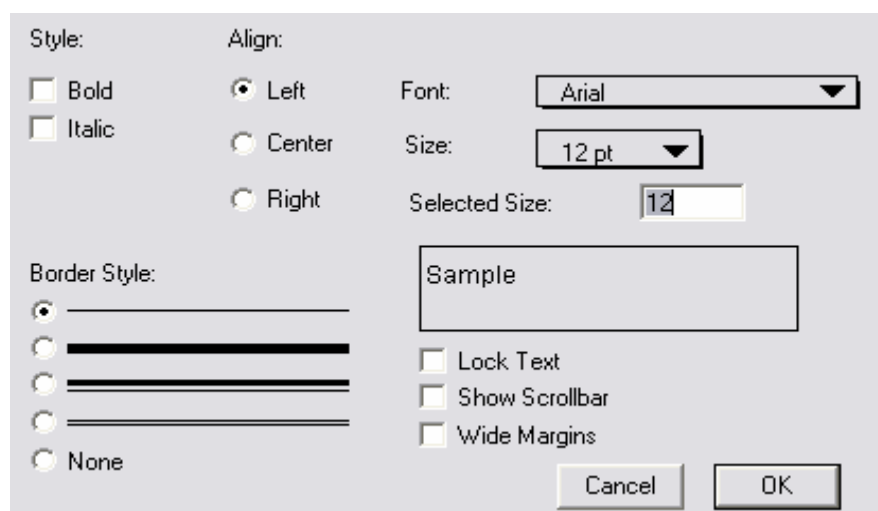


Рис.2.61

В этом окне можно выбрать стиль текста, его выравнивание, шрифт, размер, стиль рамки. Флажок **Lock Text** позволяет запретить изменения текста. Флажок **Show Scrollbar** делает видимой полосу прокрутки. Если отмечен флажок **Wide Margins**, то между рамкой и текстом существуют поля.

### 2.5.3. Рисунок



Объект **рисунок** придает модели некоторую законченность. С помощью него на уровне интерфейса или модели можно разместить рисунки или анимации. Чтобы поместить объект на рабочий лист, надо нажать кнопку  и щелкнуть в том месте рабочего листа, где вы хотите поместить **рисунок**. Внешний вид **рисунка** показан на Рис.2.62.



Рис.2.62

Заполнить диалоговое окно не составляет большого труда. В нем можно выбрать тип объекта (просто графика или 3D), написать некоторые комментарии, выбрать стиль рамки и задать рисунок, который вы хотите вставить.

#### 2.5.4. Числовой дисплей

**Числовой дисплей** используется, когда надо показать текущее значение какой-либо модельной переменной. Чтобы создать **числовой дисплей**, надо нажать на кнопку , и щелкнуть в том месте рабочего листа, где вы хотите поместить объект. Диалоговое окно определения **числового дисплея** представлено на Рис.2.63.

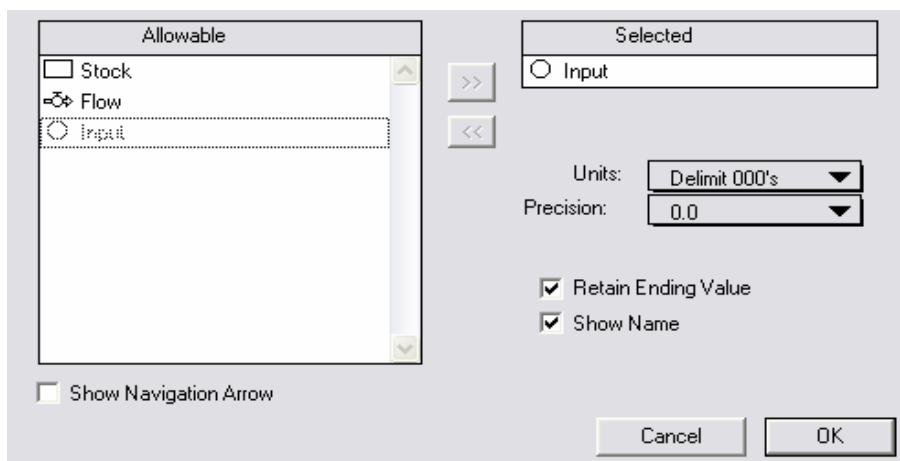


Рис.2.63

Из списка **Allowable** надо выбрать тот блок, чье значение вы хотите отобразить на дисплее, нажать кнопку >>, и выбранный элемент появится в списке **Selected**. В раскрывающихся списках **Units** и **Precision** можно выбрать единицы измерения и формат числа соответственно. Флажок **Retain Ending Value** позволяет видеть значение переменной и после конца симуляции, если флажок не поставлен, то после окончания симуляции **числовой дисплей** будет очищен. Если не отметить флажок **Show Name**, то название переменной будет не видно. Если отметить флажок **Show Navigation Arrow**, то на **числовом дисплее** появится стрелка, с помощью которой можно перейти к соответствующей переменной на модельном уровне (если **числовой дисплей** расположен на уровне интерфейса).

### 2.5.5. График


Для того чтобы создать **график**, надо нажать на кнопку  и щелкнуть в том месте рабочего листа, где вы хотите его разместить. На схеме появится небольшая иконка с изображением **графика** и его названием (Рис.2.64).



Рис.2.64

Иконку можно переместить в любое место схемы модели, кроме того, можно изменить название **графика**. Чтобы увидеть изображение **графика**, надо два раза щелкнуть левой кнопкой мыши по этой иконке. Окно **графика** показано на Рис.2.65.

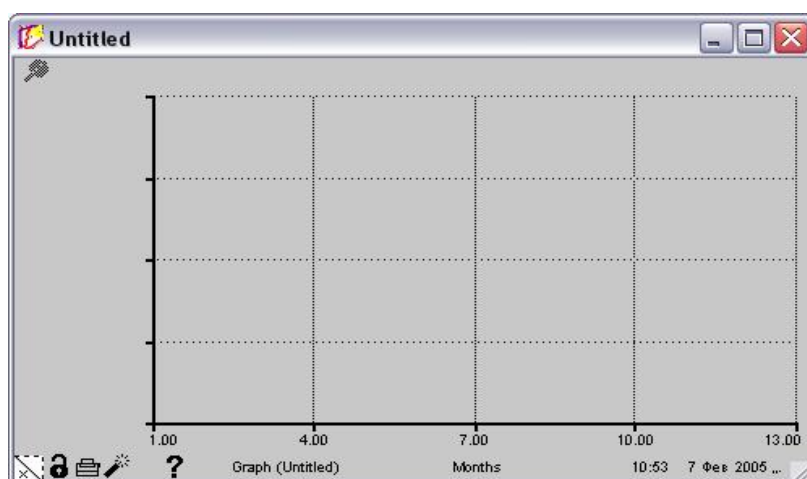


Рис.2.65

В строке заголовка указывается название **графика**, но не то, которое написано под иконкой. Если вы нажмете кнопку закрыть в строке заголовка, то окно графика будет закрыто, но в модели останется иконка. В левом верхнем углу находится кнопка с изображением кнопки, которая позволяет прикрепить **график** к рабочему листу. В этом случае он будет виден при любых перемещениях между уровнями и элементами модели. В нижней части окна **графика** находится так называемая **строка состояния графика**. В

ней указано дата и время последнего запуска модели, название оси X, название **графика**. В левом нижнем углу находится несколько кнопок. В белом квадрате указывается номер страницы, если страница одна то стоит x. Замок позволяет запретить доступ к изменениям в **графике**. Следующая кнопка с изображением принтера, дает возможность напечатать **график**. Динамит очищает содержимое **графика**. Знак вопроса вызывает окно, в котором можно задать некоторые комментарии к **графику**. Если два раза щелкнуть мышкой по окну **графика**, то появляется диалоговое окно определения **графика** (Рис.2.66).

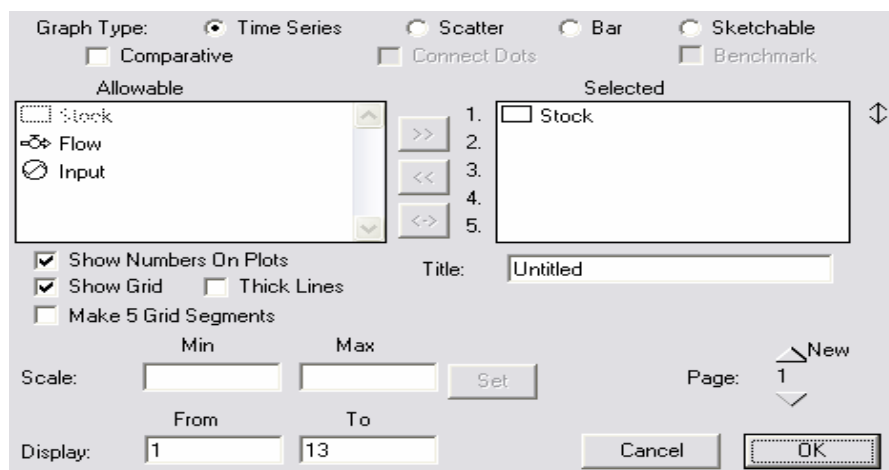


Рис.2.66

В верхней строке **Graph Type** с помощью переключателя задается тип **графика**:

- **Time Series** – это тип **графика**, который установлен по умолчанию. Он представляет собой временной ряд, то есть зависимость выбранной переменной от модельного времени.

- **Scatter** – тип **графика**, который отражает зависимость значений одной переменной от другой. В этом случае можно выбрать только две переменных, которые и будут соответствовать осям. При данном типе **графика** становится активным флажок **Connect Dots**, выбрав который точки на **графике**, отображающие зависимость переменных, будут соединены линиями.

- **Bar graphs** – тип **графика**, который представляет переменную в виде столбчатой диаграммы. Если вы хотите сравнить какие-либо значения, то столбчатая диаграмма успешно справится с этой задачей.

- **Sketchable** – позволяет создать так называемый эскиз **графика**, то есть ожидаемое значение переменной в сравнении с настоящим значением после

симуляции. В этом случае становится активным флажок **Benchmark**, который позволяет пользователям перерисовывать эскиз целиком в любой момент времени в течении симуляции.

Флажок **Comparative** часто используется в сочетании с анализом чувствительности. Если флажок отмечен, то **график** будет сравнительным, и, при нажатии на «?» в левом нижнем углу окна графика, появится ведомость, в которой указаны последние параметры анализа чувствительности, установленные при описании **графика**.

При помощи списков **Allowable** и **Selected** можно из всех доступных переменных выбрать те переменные, которые будут отображаться на **графике**. С помощью кнопки >> переменная перемещается из списка **Allowable** в список **Selected**; с помощью кнопки << из списка **Selected** в список **Allowable**, и с помощью кнопки <-> две переменные из разных списков меняются местами.

Опции графика определяются флажками:

- ü **Show Numbers On Plots** – показывает на линии **графика** его номер.
- ü **Show Grid** – показывает сетку на **графике**.
- ü **Make 5 Grid Segments** – делит сетку на 5 частей по оси X. Если флажок не отмечен, то ось разделена на 4 части.
- ü **Thick Lines** – если отметить это флажок, то **график** будет нарисован жирной линией.

В поле **Title** можно ввести **заголовок графика**. В поле **Display** устанавливаются начало и конец оси X. Надо отметить, что значение, указанное в поле **To**, не должно превышать длительности симуляции. С помощью стрелок вверх и вниз рядом с надписью **Page** можно перемещаться между страницами или создавать новые страницы.

С правой стороны от списка **Selected** стоят значки (Рис.2.67). Рассмотрим, что они обозначают.

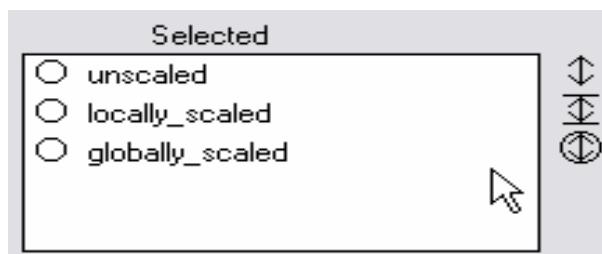


Рис.2.67

С помощью этого значка вы можете определить вид масштабирования переменных: не масштабированная переменная, переменная с локальным масштабом и переменная с глобальным масштабом. Каждому виду масштабирования соответствует свой тип значка. Тип значка можно изменить, щелкая по нему мышкой (при этом должна быть выбрана соответствующая переменная в списке).

Если вы хотите узнать координаты какой-либо точки на **графике**, то надо нажать левую кнопку мыши в этой точке. Около осей появятся координаты точки, а на **графике** вертикальная прямая, проходящая через точку. Удерживая нажатой левую кнопку мыши, прямую можно перемещать, определяя координаты других точек на **графике**. Если вы два раза щелкните на названии переменной в окне **графика**, то появится диалоговое окно, в котором можно установить числовой формат этой переменной.

### 2.5.6. Таблица


Объект **таблица** позволяет вывести числовые результаты моделирования в виде таблицы. Для того чтобы создать **таблицу**, нажмите на кнопку , а потом щелкните по рабочему листу. На схеме модели появится иконка (Рис.2.68).



Рис.2.68

Эту иконку можно перемещать по рабочему листу, изменять название. Для того чтобы увидеть непосредственно **таблицу**, надо два раза щелкнуть мышкой по этой иконке. Внешний вид таблицы показан на Рис.2.69.

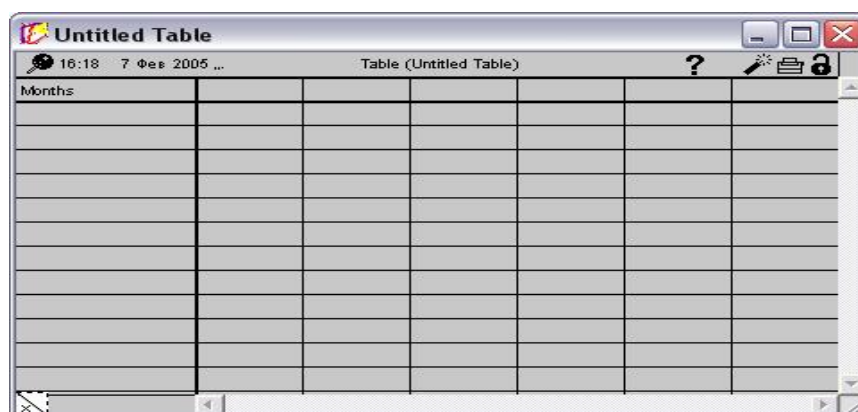




Рис.2.69

Строка состояния **таблицы** аналогична строке состояния **графика**, но находится над таблицей. В ней также указано название **таблицы**, дата и время последнего изменения модели. С помощью кнопок в верхнем правом углу строки состояния находятся кнопки, с помощью которых можно прочесть заметки к **таблице**, очистить ее содержимое, распечатать **таблицу** и закрыть доступ к ее изменению. Чтобы изменить ширину столбцов, надо нажать мышкой на границу столбца и перенести ее на нужное расстояние. Если два раза щелкнуть на **таблице**, то появляется диалоговое окно определения **таблицы** (Рис.2.70):

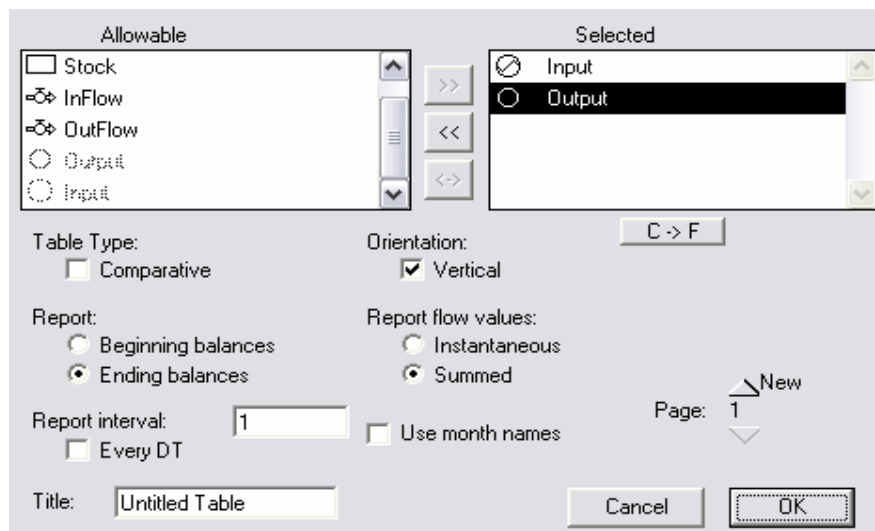


Рис.2.70

При помощи списков **Allowable** и **Selected** можно из всех доступных переменных выбрать те переменные, которые будут отображаться в **таблице**. С помощью кнопки >> переменная перемещается из списка **Allowable** в список **Selected**; с помощью кнопки << из списка **Selected** в список **Allowable**, и с помощью кнопки <-> две переменные из разных списков меняются местами.

Флажок **Comparative** устанавливает, что **таблица** является сравнительной. Это позволяет сопоставить между собой результаты нескольких. Флажок **Vertical** устанавливает вертикальную ориентацию **таблицы**.

Кнопка **C->F** позволяет установить соответствие между значением потока и значением конвертера. Программа работает таким образом, что, даже если величина потока приравнена к величине конвертера, то после работы модели в **таблице** могут быть выведены другие числа. Если зависимость установлена, то около названия блока появляется звездочка.

Переключатели **Beginning balances** и **Ending balances** позволяют прикрепить значение переменной к началу или концу временного промежутка соответственно. Флажки **Instantaneous** и **Summed** позволяют определить тип значения потока в **таблице**: мгновенный (значение потока на какой-либо момент времени) или агрегированный (суммированный за весь период). Эти значения будут отличаться, если величина промежутка между сообщением данных (значение, задаваемое в поле **Report interval**) больше величины **DT**. Если флажок **Every DT** отмечен, то данные передаются в **таблицу** через каждый интервал времени **DT**. Флажок **Use month names** позволяет вместо чисел в первом столбце увидеть названия месяцев (в том случае, если в качестве единиц измерения времени выбраны месяцы). В поле **Title** задается **название таблицы**. С помощью стрелок вверх и вниз рядом с надписью **Page** можно перемещаться между страницами или создавать новые страницы.

Если два раза щелкнуть левой кнопкой мыши на заголовке столбца, то появится диалоговое окно, в котором можно определить стиль и выравнивание заголовка, единицы измерения, масштаб и формат чисел в **таблице**.

## **ГЛАВА 3. ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ АНАЛИЗА ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ**

### **3.1. Влияние изменения процентной ставки на сальдо банковского счета**

Этот пример [11] демонстрирует простейшую модель годового роста сальдо банковского счета, скорость которого зависит от колеблющейся ежемесячной процентной ставки. Хорошо известно, что если после очередного интервала начисления доход (т. е. начисленные за данный интервал проценты) не выплачиваются, а присоединяются к денежной сумме, имеющейся на начало этого интервала, для определения наращенной суммы применяют формулы сложных процентов. Такие сложные ссудные проценты в настоящее время являются весьма распространенным видом процентных ставок.

Для начала создаем схему модели на модельном уровне. Размещаем на рабочем листе **резервуар**, **поток** и **конвертер**. Присваиваем им соответствующие названия. Кроме того, помещаем на рабочую поверхность объекты **график** и **таблица**, в которых будут отображаться результаты работы модели (Рис.3.1).

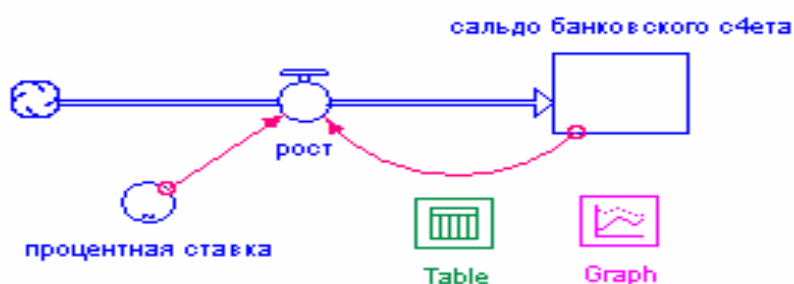


Рис.3.1

Отметим, что при использовании русских букв в названии блоков некоторые буквы, например «ч» могут не прописываться, поэтому мы заменили ее «4».

Теперь переходим на этап численного описания, нажимая на кнопку с изображением **земного шара** слева от рабочего листа. На каждом элементе схемы появляется изображение **знака вопроса**. Двойной щелчок по конвертеру открывает диалоговое окно определения числовых параметров конвертера. В поле инициализации указываем, что мы будем использовать функцию **TIME**, то есть выбираем ее из списка **Builtins**. И нажимаем на кнопку **Become Graphical Function**. Таким образом, мы зададим процентную ставку в зависимости от модельного времени (месяца). Появляется диалоговое окно определения графической функции (Рис.3.2):

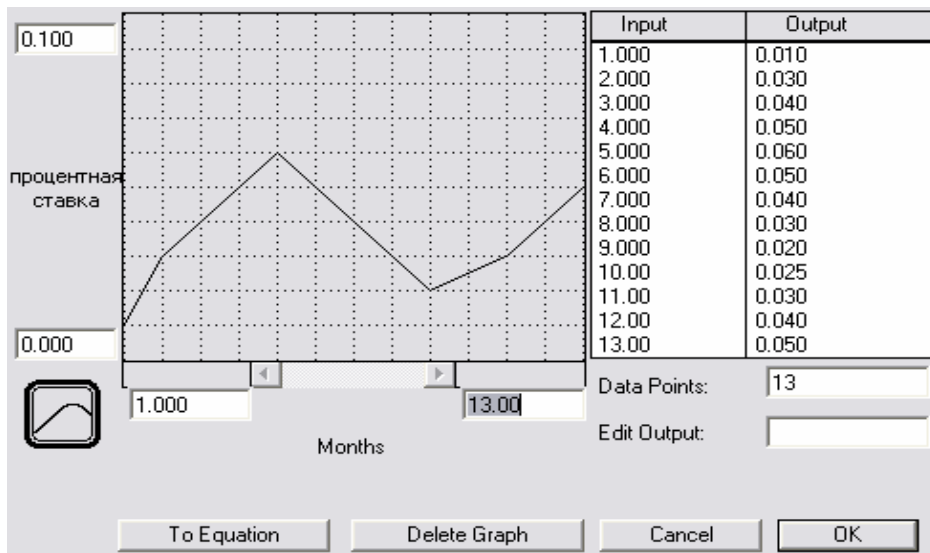


Рис.3.2

Указываем значения нашей процентной ставки и нажимаем **ОК**.

После двойного щелчка по резервуару появится диалоговое окно (Рис.3.3), в котором мы зададим начальные значения сальдо банковского счета 10000 и зададим единицы измерения: ЕВРО (для этого нажать на кнопку **Units** и в открывшемся окне из выпадающего списка выберем **Euro**).

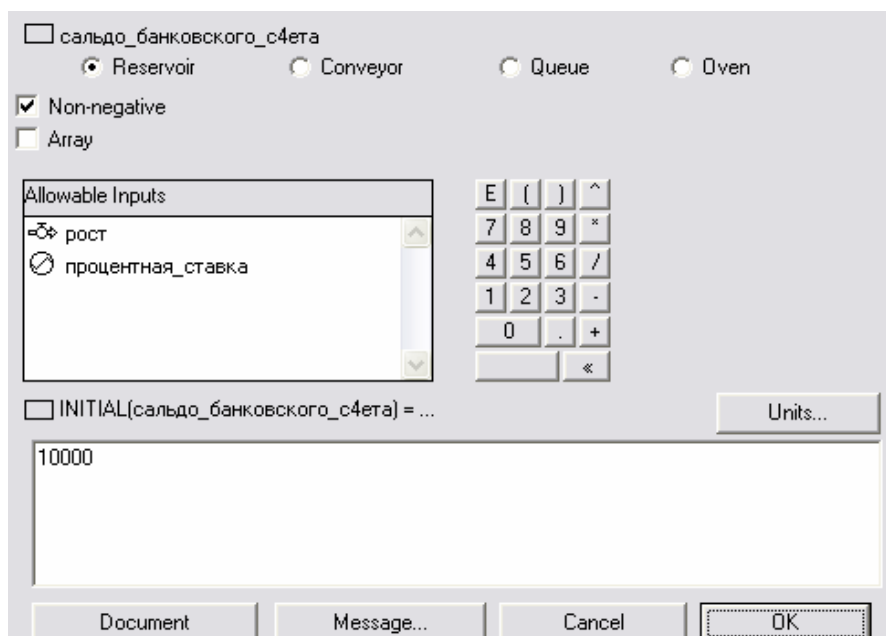


Рис.3.3

После этого откроем диалоговое окно определения числовых параметров потока. В нем мы должны задать формулу роста сальдо при помощи переменных из списка **Required Inputs** и калькулятора (Рис.3.4).

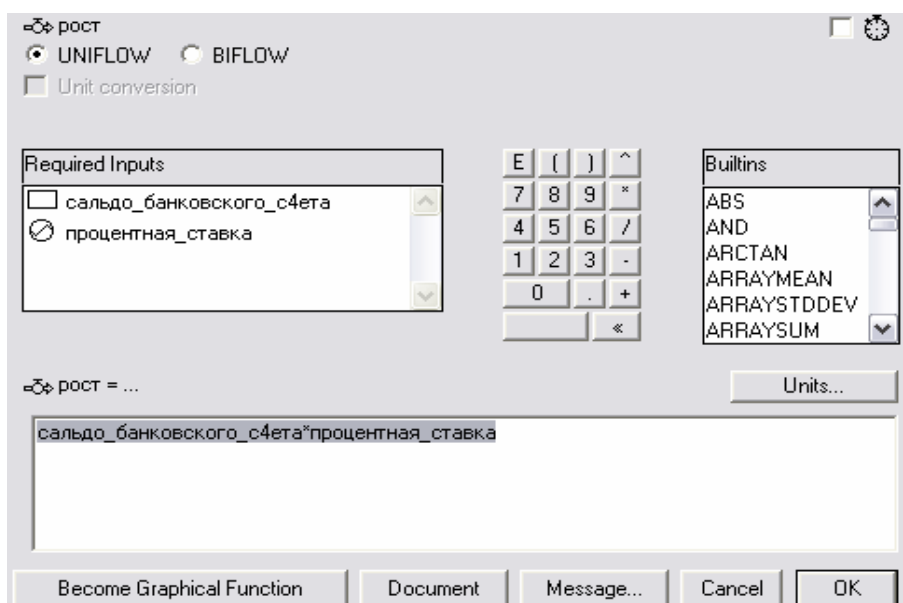


Рис.3.4

Теперь, когда мы определили все переменные модели, можно определить **график**. Два раза щелкаем мышкой по иконке графика, появляется окно графика. После двойного щелчка по нему появляется диалоговое окно определения графика, в котором мы выбираем переменные, которые будем рисовать на графике (Рис.3.5). Это выполняется с помощью кнопок >>, << и <->. Кроме того, укажем название графика.

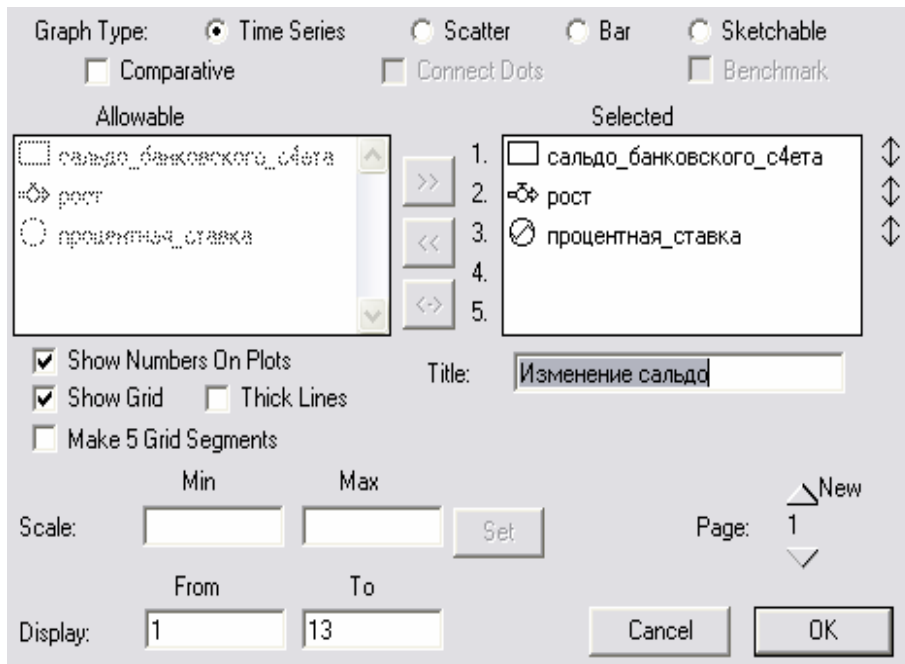


Рис.3.5

Теперь откроем диалоговое окно определения таблицы (Рис.3.6). В нем выберем те переменные, значения которых мы будем иллюстрировать таблицей, и зададим остальные параметры нашей таблицы.

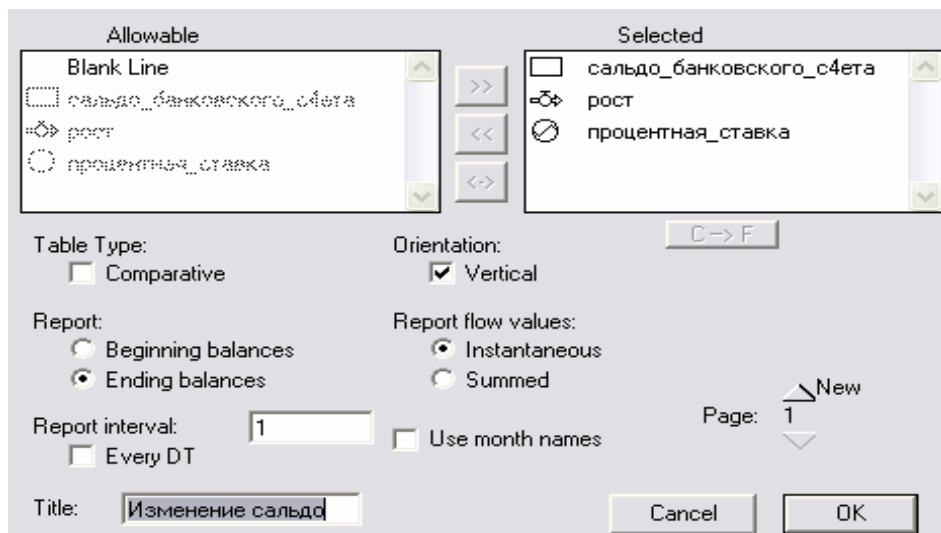


Рис.3.6

Надо заметить, что программный код сгенерируется автоматически. Если перейти на уровень программного кода с помощью стрелки вниз слева от рабочего листа, то можно увидеть текст программы.

```
□ сальдо_банковского_счета(t) = сальдо_банковского_счета(t - dt) + (рост) * dt
  INIT сальдо_банковского_счета = 10000
  INFLOWS:
    ↻ рост = сальдо_банковского_счета*процентная_ставка
  ○ процентная_ставка = GRAPH(TIME)
    (1.00, 0.01), (2.00, 0.03), (3.00, 0.04), (4.00, 0.05), (5.00, 0.06), (6.00, 0.05), (7.00, 0.04), (8.00, 0.03),
    (9.00, 0.02), (10.0, 0.025), (11.0, 0.03), (12.0, 0.04), (13.0, 0.05)
```

После определения всех параметров модели выполним ее запуск. Для этого выберем из меню **Run** пункт **Run**. Результат работы модели будет отражен в таблице и на графике (Рис.3.7). Для пояснения сути модели мы разместили на рабочем листе объект текст, в котором написали название модели.

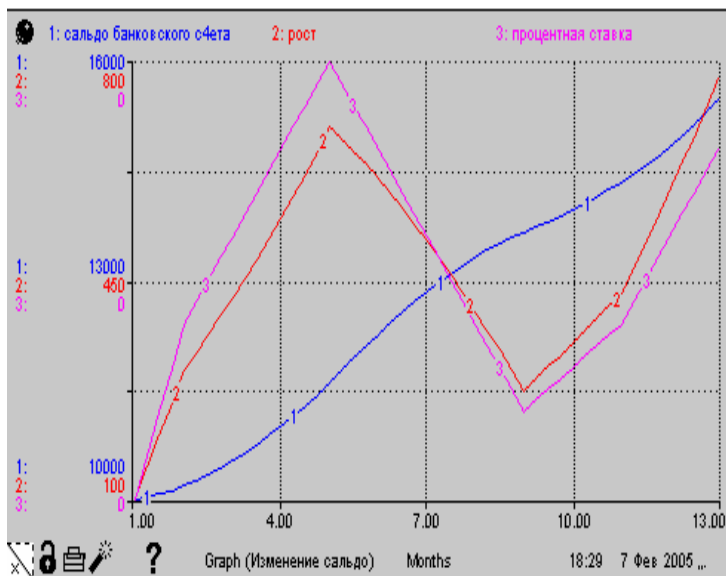
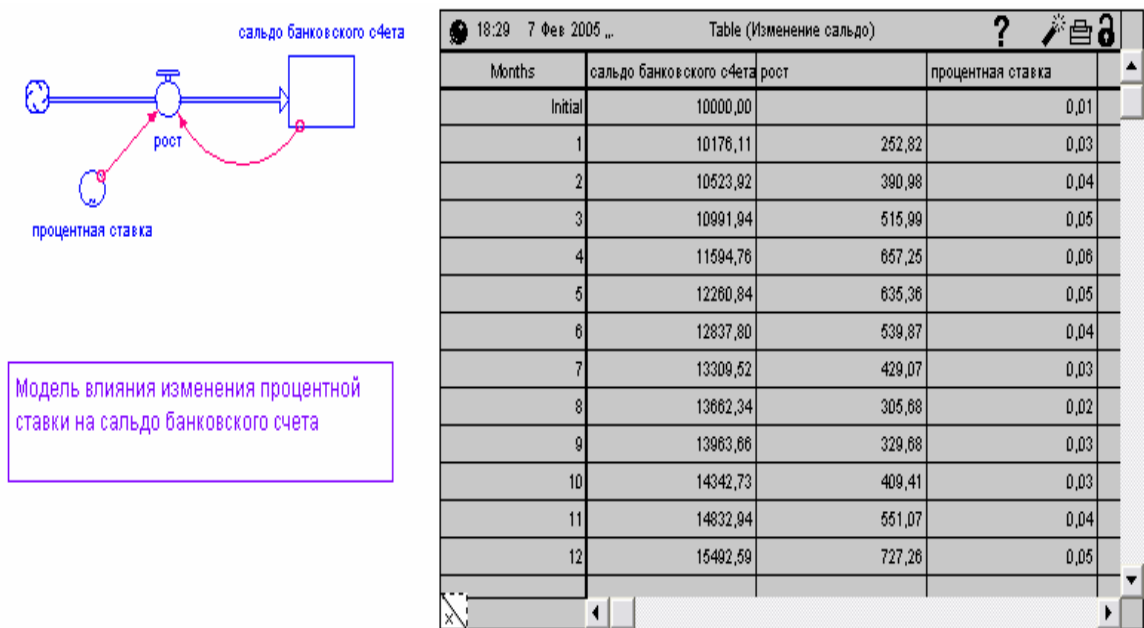


Рис.3.7

### 3.2. Модель распределения доходов фирмы между доходами от продажи и сервисным обслуживанием

Рассмотрим модель [11], демонстрирующую целесообразность и перспективность ориентации фирм, производящих компьютерное оборудование, на последующую концентрацию своих усилий в сфере оказания сервисных услуг по сопровождению. Модель весьма универсальна и может использоваться в самых разнообразных областях применения потоковых схем. Приведем схему модели (Рис.3.8).



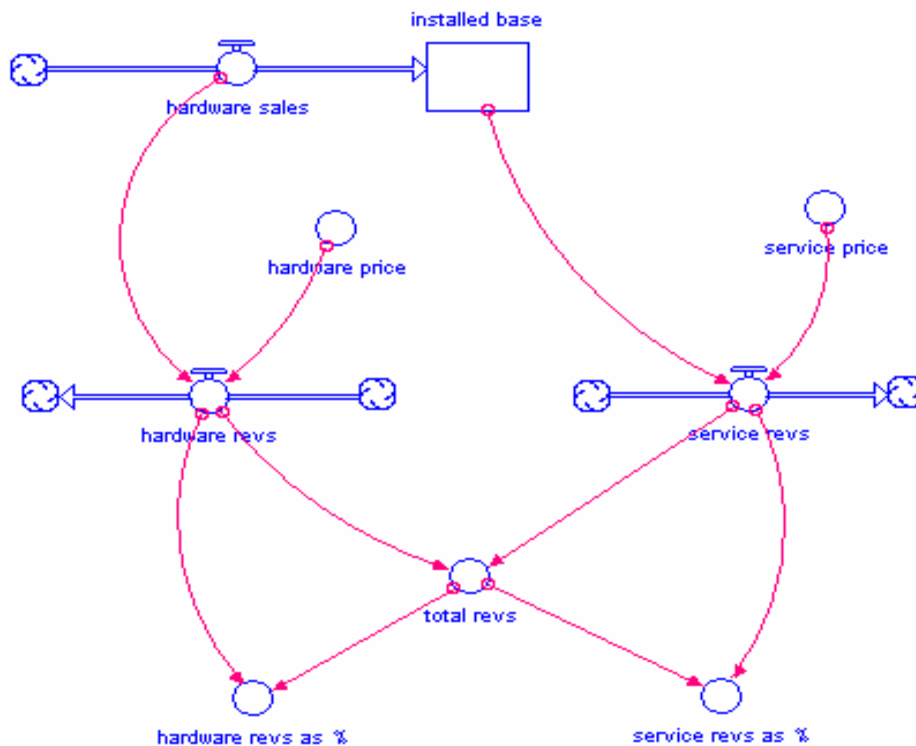


Рис.3.8

Модель концептуально возникла после исследования процессов, происходящих в компьютерном бизнесе после бума 60-х, когда в США произошло насыщение рынка вычислительным оборудованием, и когда в условиях жесточайшей конкуренции некоторые динамично развивающиеся фирмы обратили внимание на новый привлекательный сектор рынка. Далее модель в уточненном виде нашла свое применение в других сферах деловой активности и в какой-то мере стала классической.

Прежде всего, потребуется поток **hardware sales**, отображающий лишь эскизно динамику возможно непростого бизнес - процесса продажи наукоемкого вычислительного оборудования и соответствующего базового системного и прикладного программного обеспечения, которому предшествует целый комплекс мероприятий и исследований маркетингового характера.

Фонд **installed base** предполагает наличие расширяющегося парка компьютеров, потенциально нуждающихся в постоянном сервисном обслуживании.

На идеограмме видно, что поток **hardware revs** зависит от потока **hardware sales** и конвертора **hardware price**, а поток **service revs** - от потока **service price** и содержимого фонда **installed base**. Интенсивности этих двух симметричных потоков управляются через вычисление значений соответствующих очевидных выражений, которые вы будете

задавать при определении параметров схемы. Очевидно, что вопросы о возможных колебаниях реальных цен никак не учитываются нашей моделью и мы, естественно, ограничимся лишь модельными предположениями.

Наконец, все доходы фирмы собираются в конвертере **total revs** и затем пропорционально разделяются на **hardware revs as a %** и **service revs as a %**. Динамика возможных соотношений между выходными значениями соответствующих конвертеров в зависимости от колебания цен на компьютеры, изменением стоимости сервиса, а также увеличением парка установленного оборудования и составляет цель имитационных экспериментов на модели.

Определение и уточнение схемы производится на фазе окончательного формирования поведенческой модели, а выход на этот режим управления осуществляется через соответствующую кнопку на левой стороне диалогового окна.

Вызываем двойным щелчком мыши соответствующие диалоговые окна, определяем через них значения констант, формируем алгебраические выражения или - задаем возможные начальные значения или соотношения через графические функции.

Зададим значение потока **hardware sales** с помощью графической функции. Диалоговое окно определения графической функции примет вид (Рис.3.9):

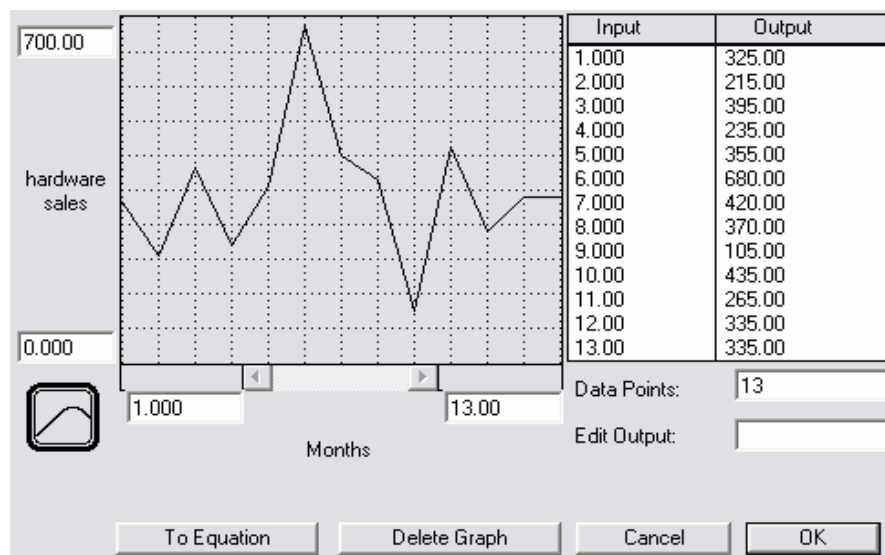


Рис.3.9

Аналогичным образом определяются числовые параметры конвертеров **hardware price** и **service price**. Определим начальное значение резервуара **installed base** (Рис.3.10), то есть начальное значение количества проданных компьютеров.



Рис.3.10

Теперь определим значение потоков **hardware revs** и **service revs**. Диалоговое окно определения числовых параметров потока имеет вид (Рис.3.11):

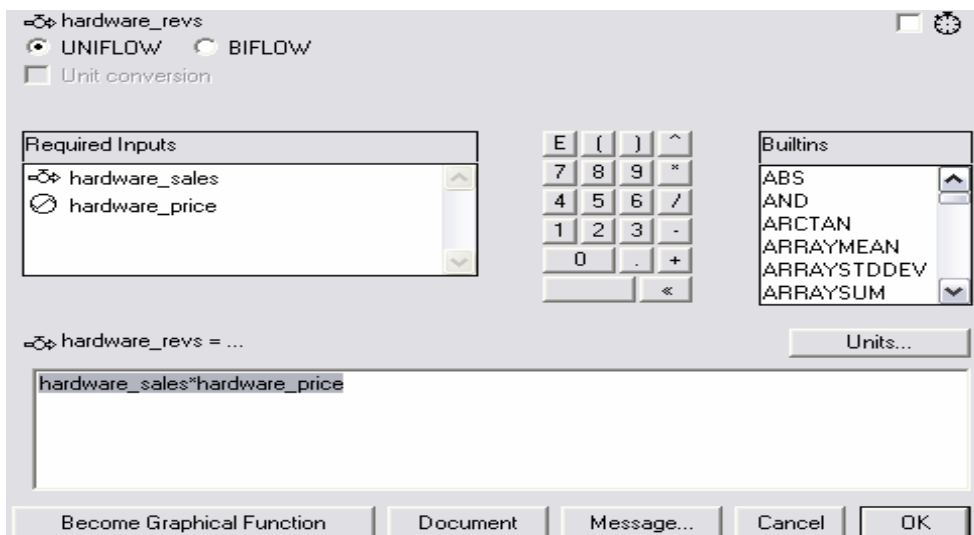


Рис.3.11

В поле инициализации задается формула при помощи выбора блоков из списка **Required Inputs** и калькулятора. Аналогичным образом задается и поток **service revs**.

Диалоговое окно определения числовых параметров конвертера **total revs** показано на Рис.3.12.

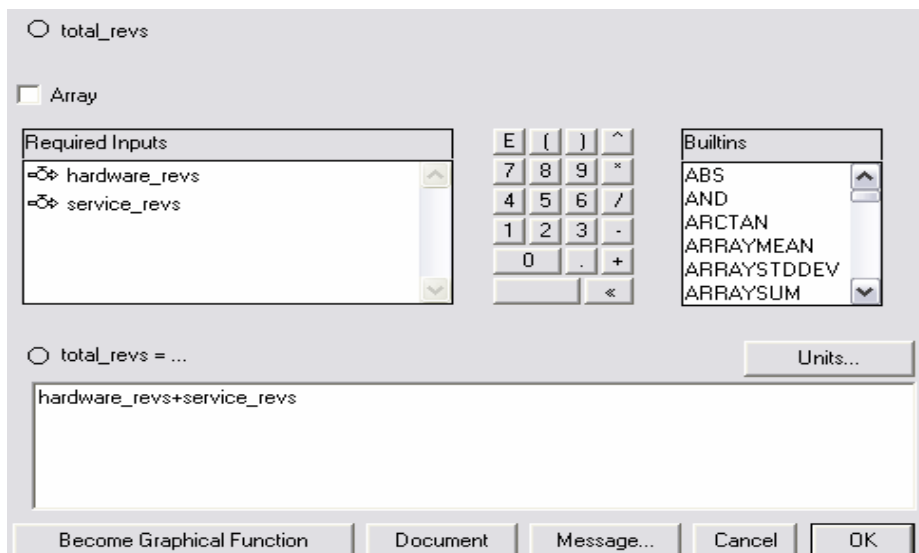


Рис.3.12

Диалоговые окна для определения конвертеров **service revs as %** и **hardware revs as %** имеют аналогичный вид.

Закончив все возможные определения элементов модели (в чем убеждаемся по отсутствию изображений фондов, потоков и конвертеров с предупреждающими знаками «?»), можно приступить к оформлению пользовательского интерфейса модели.

Для того чтобы увидеть результаты, добавим в модель объект **график**. Его свойства определяются при помощи диалогового окна, изображенного на Рис.3.13:

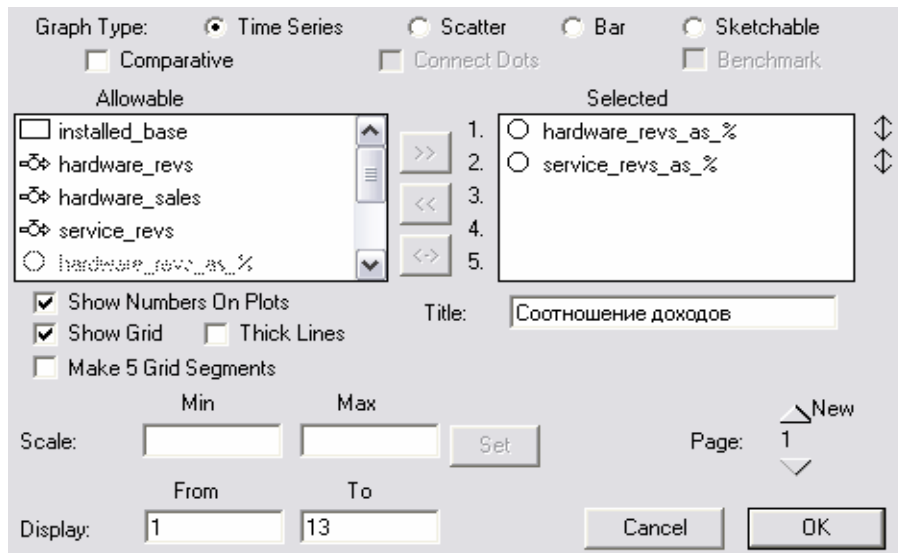


Рис.3.13

Добавим в модель **таблицу**, в которой будут отражаться изменения всех элементов модели. Диалоговое окно определения объекта таблица выглядит следующим образом (Рис. 3.14):

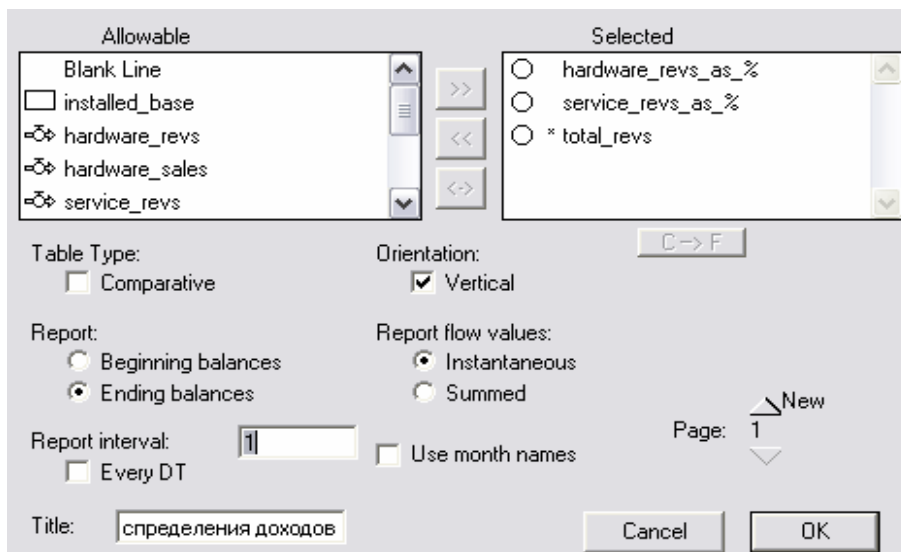


Рис.3.14

Установив соответствующие настройки (или используя готовые по умолчанию) для проведения имитационных экспериментов, через меню **Run** запускаем режим имитации и,

отслеживая продвижение процесса в интервале задаваемого модельного времени через индикатор, ждем результат.

На Рис. 3.15 приведен результат имитационного эксперимента, демонстрирующий общую картину поведения модельных единиц.

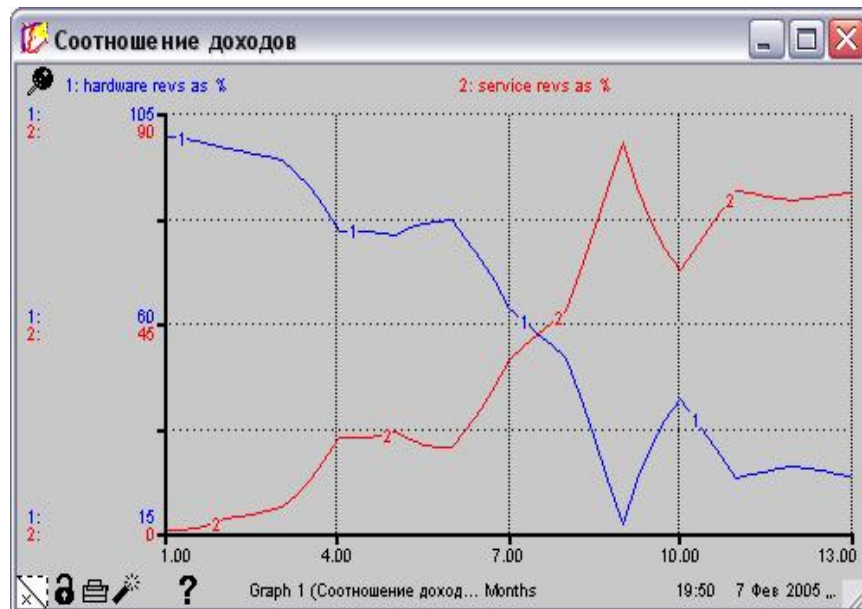


Рис.3.15

После запуска таблица будет выглядеть следующим образом (Рис. 3.16):

Months	hardware revs as %	service revs as %	total revs
1	97,40	2,60	1134043,20
2	94,89	5,11	1446396,88
3	79,66	20,34	1166859,38
4	78,70	21,30	1270468,75
5	81,94	18,06	2172399,22
6	63,13	36,87	1968445,31
7	52,41	47,59	1774853,13
8	16,41	83,59	1500022,66
9	43,53	56,47	2062324,22
10	26,53	73,47	2088639,06
11	28,91	71,09	2219828,91
12	26,96	73,04	2389343,75

Рис.3.16

Большие и сложные схемы можно декомпозировать на подсхемы и инкапсулировать их в соответствующие фреймы секторов, как это продемонстрировано на Рис.3.17.

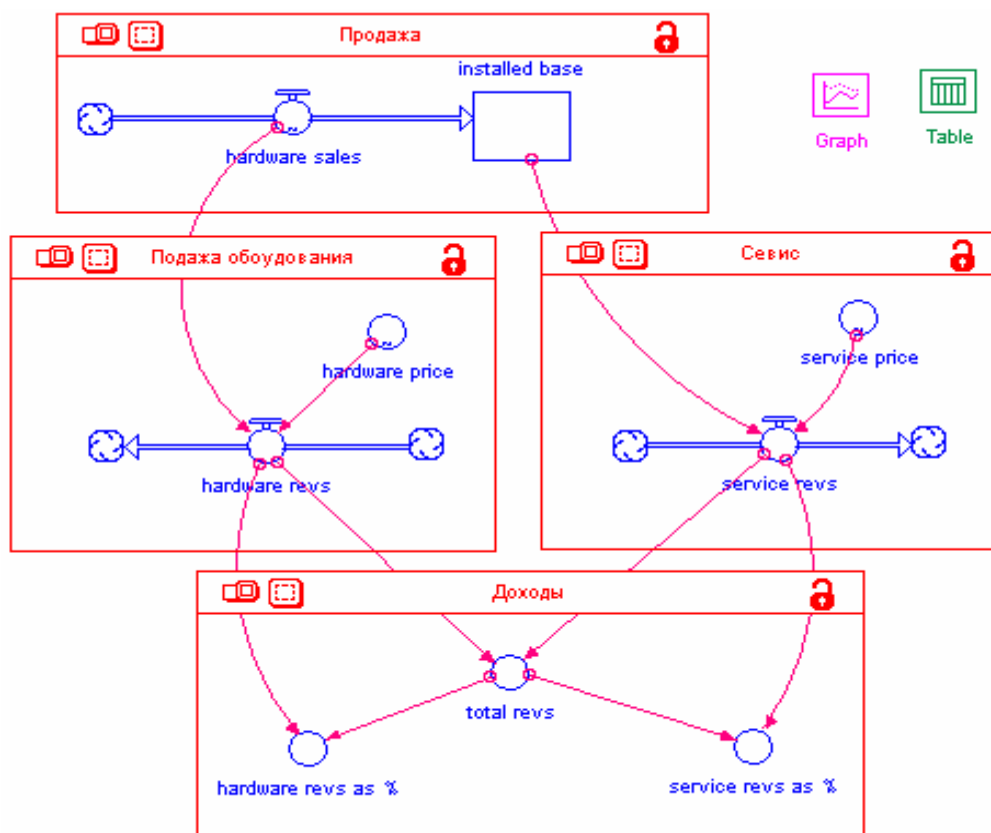


Рис.3.17

Если после этого поднимемся с помощью стрелки на правой панели управляющего окна вверх, то окажемся на самом верхнем уровне представления модели, где **Ithink** предложит отображение фреймов процессов в виде традиционных блок-схем, при условии, что в пункте меню **Interface Prefs** на уровне интерфейса поставлен флажок **Link High-Level Map to Model** (Рис.3.18).

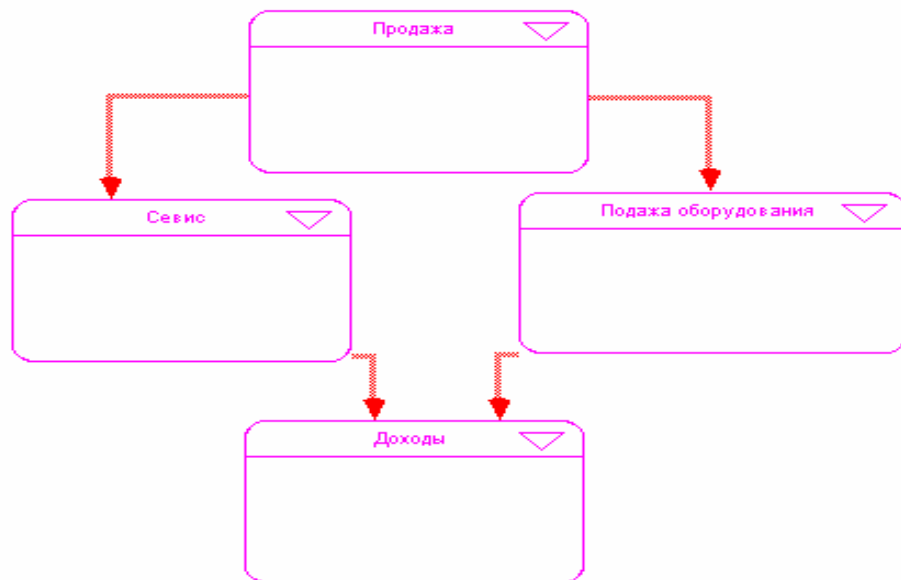


Рис.3.18

Наконец, если у нас появилось желание увидеть программный код модели, который автоматически сформировал **Ithink**, опускаемся с помощью стрелки на правой панели управляющего окна вниз. Мы окажемся на самом нижнем уровне представления модели и увидим в окне следующий текст:



#### Доходы

- $\text{hardware\_revs\_as\_}\% = 100 * \text{hardware\_revs} / \text{total\_revs}$
- $\text{service\_revs\_as\_}\% = 100 * \text{service\_revs} / \text{total\_revs}$
- $\text{total\_revs} = \text{hardware\_revs} + \text{service\_revs}$

#### Подажа оборудования

UNATTACHED:

- $\text{hardware\_revs} = \text{hardware\_sales} * \text{hardware\_price}$
- $\text{hardware\_price} = \text{GRAPH}(\text{TIME})$   
(1.00, 4700), (2.00, 4575), (3.00, 3750), (4.00, 3550), (5.00, 2950), (6.00, 2975), (7.00, 2725), (8.00, 2500), (9.00, 2225), (10.0, 2300), (11.0, 2025), (12.0, 1975), (13.0, 1950)

#### Продажа

- $\text{installed\_base}(t) = \text{installed\_base}(t - dt) + (\text{hardware\_sales}) * dt$

INIT installed\_base = 25

INFLOWS:

- $\text{hardware\_sales} = \text{GRAPH}(\text{TIME})$   
(1.00, 325), (2.00, 215), (3.00, 395), (4.00, 235), (5.00, 355), (6.00, 680), (7.00, 420), (8.00, 370), (9.00, 105), (10.0, 435), (11.0, 265), (12.0, 335), (13.0, 335)

#### Севис

UNATTACHED:

- $\text{service\_revs} = \text{installed\_base} * \text{service\_price}$
- $\text{service\_price} = \text{GRAPH}(\text{TIME})$   
(1.00, 19.5), (2.00, 85.0), (3.00, 135), (4.00, 230), (5.00, 235), (6.00, 265), (7.00, 295), (8.00, 315), (9.00, 405), (10.0, 410), (11.0, 420), (12.0, 425), (13.0, 425)

#### Not in a sector

Отметим, что существенным свойством потоковых моделей является весьма высокий уровень их абстрагирования, что делает возможным достаточно простую адаптацию моделей к другим областям применения, а зачастую и весьма неожиданную их интерпретацию.

### 3.3. Модель кредитования трех предприятий

Для банка типична ситуация, когда у него имеется группа приоритетных клиентов. Это могут быть традиционные клиенты банка или промышленные предприятия, входящие в состав его финансово-промышленной группы. При планировании работы с этой группой клиентов встает вопрос о распределении кредитных ресурсов между ними. При этом руководство должно определить параметры кредитования – сроки, график, ставку процента и другие условия. Решение таких задач обычно требует перебора большого числа различных вариантов. Они могут решаться при помощи специализированных банковских программных средств, имеющих более или менее жесткий интерфейс. Обычно такие программы являются весьма дорогостоящими и узконаправленными. В отличие от них система **Ithink** обладает большей гибкостью, и с ее помощью могут быть решены как аналогичные, так и совершенно другие задачи.

Итак, каким образом можно решить задачу о распределении кредитных ресурсов с помощью пакета **Ithink**? Прежде всего, необходимо отметить, что это типично «поточковая» задача. Поток денежных средств поступает из банка на предприятие, удерживается в течение некоторого времени, а затем либо выводится из модели, либо возвращается в банк. Основной проблемой является разработка определенной структуры отношений между клиентами – очередность, приоритетность и определение основных условий. При изменении этой структуры клиенты получают уже иные кредиты в иные сроки. Общий размер выделенных кредитных ресурсов также сказывается на условиях кредитования отдельных клиентов. Пакет **Ithink** позволяет оператору изменять приоритеты и условия кредитования, в результате изменяются и графики кредитования предприятий, и динамика состояния расчетного счета банка. Все это позволит аналитику быстро просмотреть различные варианты кредитных схем и выбрать среди них оптимальную.

Для простоты заложим в модель следующие приоритеты [12]. Предприятие №1 – главный партнер банка. Оно получает кредиты в первую очередь. Если кредитные возможности банка превышают потребности предприятия (10 млн. руб.), то ресурсы направляются предприятию №2. Максимальный размер кредита, который может получить предприятие №2, также ограничен 10 млн. руб. Избыток направляется предприятию №3. Предел по размеру кредита для третьего предприятия составляет 5 млн. руб. Серия кредитов представлена четырьмя кредитами по 15 млн. руб., поступающими ежеквартально. Первому предприятию кредит может выдаваться на 3 квартала, второму – только на 2 квартала, третьему – на 1 квартал.

Кроме того, предполагаем, что денежные средства, «прошедшие» через предприятия, выводятся из процесса.

В данной задаче речь идет о некоторой последовательности – потоке кредитов. В пакете **Ithink** существует элемент **конвейер** – полный аналог предприятия, принимающего кредит. Денежная сумма поступает в него, удерживается в течение определенного срока, а затем возвращается в общий поток. Элемент **очередь** моделирует серию кредитов, то есть исполняет в данной модели функции банка. В нем устанавливается приоритетность распределения кредитных ресурсов. С помощью **потоков** денежные средства поступают на предприятия. При помощи структурных элементов пакета **Ithink** мы создаем модель процесса кредитования на базовом уровне представления потоковыми схемами. Модель представлена на Рис.3.19.

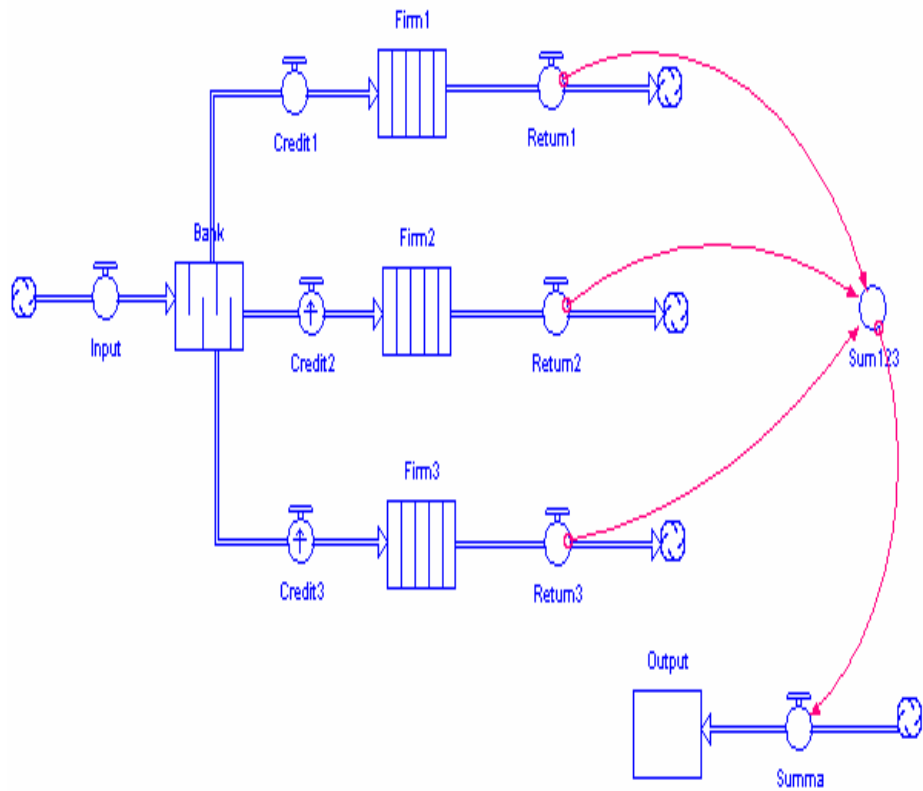


Рис.3.19

Теперь мы должны ввести в нашу модель все числовые данные, для этого мы в окне **Ithink** нажимаем кнопку с изображением **маркера в виде глобуса**. На нашей модели появляются **знаки вопроса** на тех блоках, параметры которых мы должны определить. Для задания числовых характеристик необходим двойной щелчок левой кнопкой мыши по определяемому объекту.

Для начала зададим параметры входного потока **INPUT** (Рис.3.20).

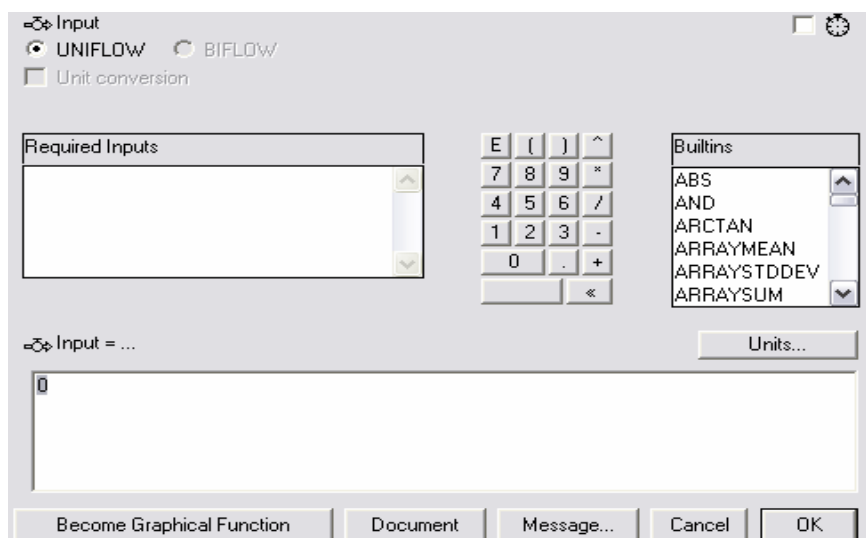


Рис.3.20

После задания параметров потока надо нажать кнопку **OK**, после чего изображение знака вопроса со значка потока исчезнет.

Теперь зададим параметры очереди. Для этого также два раза щелкаем левой кнопкой мыши по структурному элементу **Bank**. Появится окно (Рис.3.21):

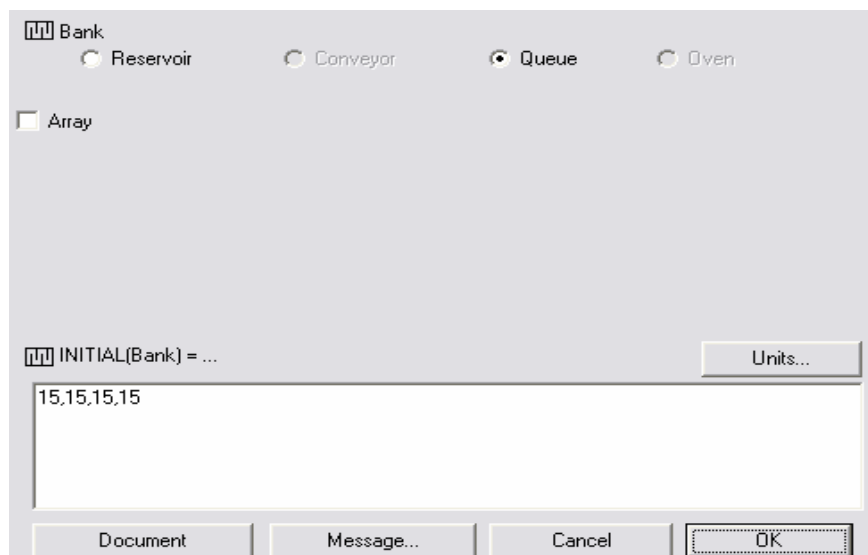


Рис.3.21

В диалоговом окне выходных потоков из очереди **Credit2** и **Credit3** ставим флажок **Overflow** для определения приоритетов выходных потоков. На изображении потока появляется **крестик**.

Определим параметры для каждого предприятия. Диалоговое окно для определения структурного элемента **конвейер** представлено на Рис.3.22.

Dialog box for 'Firm1' configuration. The 'Conveyor' option is selected. Parameters include: Transit time: 3; Inflow limit: 10; Capacity: 10; Accept: As much as possible. The initial value is set to 0.

Рис.3.22

Здесь мы определяем время транзита, **ограничения на входной поток и емкость конвейера, а также начальное значение блока**. Аналогичным образом определяются и конвейеры **Firm2** и **Firm3**.

Теперь рассмотрим инициализацию элемента **SUM\_123**. Диалоговое окно представлено на Рис.3.23. С помощью элемента **конвертер** мы должны суммировать значения потоков. Для этого мы из списка **Required Inputs** выбираем необходимые элементы и складываем их.

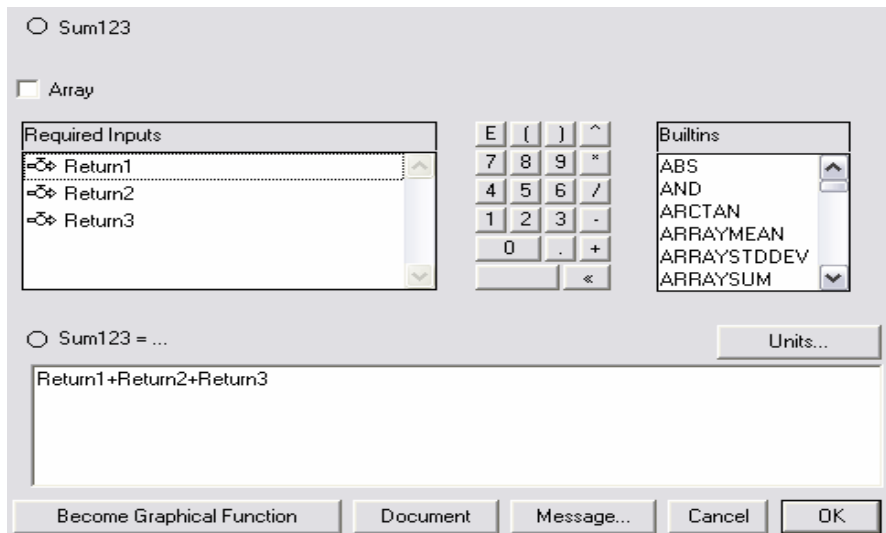


Рис.3.23

Аналогично мы определяем оставшиеся элементы модели.

В результате наших действий на программном уровне сформировался программный код. Он будет иметь следующий вид:

$$\text{BANK}(t) = \text{BANK}(t - dt) + (\text{INPUT} - \text{Credit}_1 - \text{Credit}_2 - \text{Credit}_3) * dt$$

INIT BANK = 15,15,15,15

INFLOWS:

$$\text{INPUT} = 0$$

OUTFLOWS:

$$\text{Credit}_1 = \text{QUEUE OUTFLOW}$$

$$\text{Credit}_2 = \text{QUEUE OUTFLOW}$$

$$\text{Credit}_3 = \text{QUEUE OUTFLOW}$$

$$\text{FIRM}_1(t) = \text{FIRM}_1(t - dt) + (\text{Credit}_1 - \text{Return}_1) * dt$$

INIT FIRM\_1 = 0

TRANSIT TIME = 3

INFLOW LIMIT = 10

CAPACITY = 10

INFLOWS:

$$\text{Credit}_1 = \text{QUEUE OUTFLOW}$$

OUTFLOWS:

$$\text{Return}_1 = \text{CONVEYOR OUTFLOW}$$

$$\text{FIRM}_2(t) = \text{FIRM}_2(t - dt) + (\text{Credit}_2 - \text{Return}_2) * dt$$

INIT FIRM\_2 = 0

TRANSIT TIME = 2

INFLOW LIMIT = 5

CAPACITY = 10

INFLOWS:

$$\text{Credit}_2 = \text{QUEUE OUTFLOW}$$

OUTFLOWS:

$$\text{Return}_2 = \text{CONVEYOR OUTFLOW}$$

$$\text{FIRM}_3(t) = \text{FIRM}_3(t - dt) + (\text{Credit}_3 - \text{Return}_3) * dt$$

INIT FIRM\_3 = 0

TRANSIT TIME = 1

INFLOW LIMIT = 5

CAPACITY = 5

INFLOWS:

$$\text{Credit}_3 = \text{QUEUE OUTFLOW}$$

OUTFLOWS:

$$\text{Return}_3 = \text{CONVEYOR OUTFLOW}$$

$$\text{OUTPUT}(t) = \text{OUTPUT}(t - dt) + (\text{Sum123}) * dt$$

INIT OUTPUT = Sum123

INFLOWS:

$$\text{Sum123} = \text{SUM}_{123}$$

$$\text{SUM}_{123} = \text{Return}_1 + \text{Return}_2 + \text{Return}_3$$

○

Чтобы увидеть результаты работы нашей модели, мы должны на базовом уровне поместить **графики** и связать их с соответствующими элементами модели.

Чтобы модель начала работать, мы должны выбрать пункт меню **Run**.

На выходе модели имеем, в частности, размер кредитных ресурсов, полученных каждым предприятием. При заданных приоритетах кредиты распределились следующим образом: предприятие N1 - 20 млн. руб., N2 - 25 млн. руб., N3 - 15 млн. руб. График возврата средств банку представлен на Рис.3.24.



Рис.3.24

При анализе результатов необходимо учитывать, что поток кредитов определяется не только их объемом, но и сроком кредитования. Для предприятия №2 кредитный поток представлен на Рис.3.25.

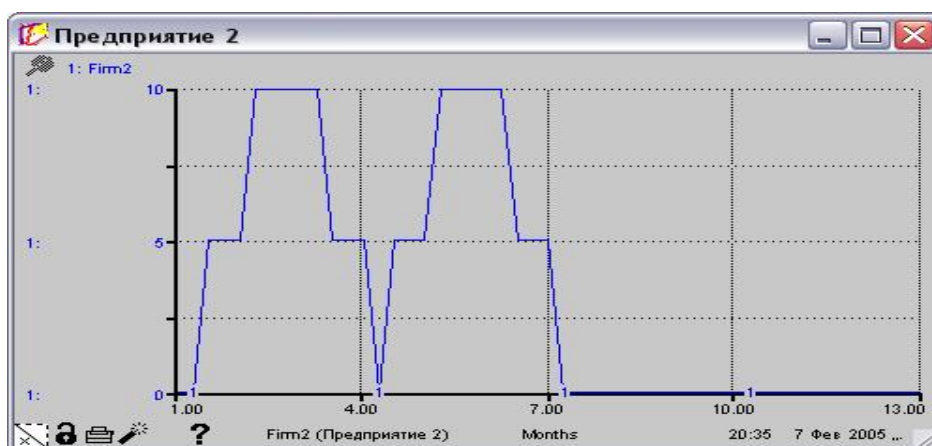


Рис.3.25

По горизонтальной оси отложено время, по вертикальной оси - количество кредитных ресурсов, находящихся в распоряжении предприятия в данный момент времени. Процесс кредитования предприятия предстает в весьма наглядном виде.

С точки зрения сроков и количества полученных ресурсов положение предприятия N 2 явно предпочтительнее третьего. Общая сумма кредитных ресурсов доходила до 25 млн. руб. (хотя и поступала меньшими порциями). У третьего предприятия сумма полученных



ресурсов составила 15 млн. руб. (без учета срочности). Кредитный поток для предприятия №3 представлен на Рис.3.26.

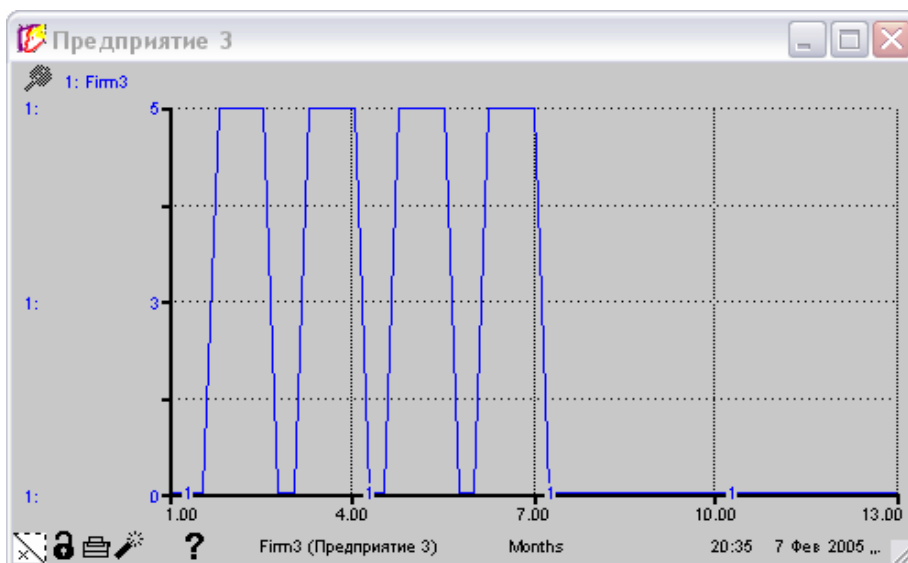


Рис.3.26

Предприятие №2 получило кредитов даже больше, чем предприятие №1. Однако с точки зрения сроков положение предприятия №1 более благоприятно. В распоряжении предприятия №1 в течение всего срока находилось не менее 10 млн. руб. (Рис.3.27).

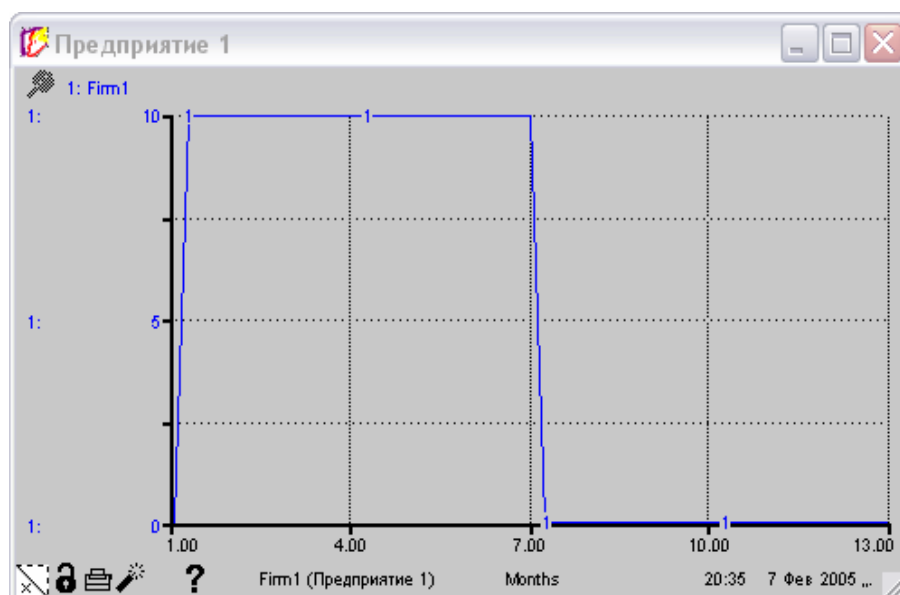


Рис.3.27

Эта модель в усовершенствованном виде, когда денежные средства вместе с процентами, которые возвращают предприятия, банк может опять выдавать в виде кредитов с учетом нормы обязательного резерва, рассмотрена в работе [10].

## ЛИТЕРАТУРА

1. Емельянов А.А. Имитационное моделирование в экономических информационных системах. / А.А. Емельянов, Е.А. Власова - М.: Изд-во МЭСИ, 1998.-108 с.
2. Емельянов А.А. Имитационное моделирование в управлении рисками.- СПб.: Инжекон, 2000.- 376 с.
3. Емельянов А.А. Имитационное моделирование экономических процессов. / А.А. Емельянов, Е.А. Власова, Р.В. Дума - М.: Финансы и статистика, 2004. – 368 с.
4. Сидоренко В.Н. Системная динамика. - М.: ТЕИС, 1998. – 205 с.
5. Sterman, J.D. System Dynamics Models for Project Management. ([www.rub.ruc.dk](http://www.rub.ruc.dk))
6. Рынок программных средств. ([www.economy.mari.ru](http://www.economy.mari.ru))
7. Кузнецов Ю.А. Работа с программным пакетом ITNINK. Учебно-методическое пособие. / Ю.А. Кузнецов, В.И. Перова, О.В. Мичасова - Нижний Новгород: Изд-во Нижегородского университета, 2005.-72 с.
8. Кузнецов Ю.А. Моделирование экономического поведения фирмы с применением программного пакета ITNINK / Ю.А. Кузнецов, В.И. Перова, О.В. Мичасова // Государственное регулирование экономики. Региональный аспект: Материалы V Международной научно-практической конференции. - Нижний Новгород: Изд-во ННГУ, 2005.- С.186-188.
9. Кузнецов Ю.А. Имитационное моделирование экономических процессов с применением программного пакета ITNINK / Ю.А. Кузнецов, В.И. Перова, О.В. Мичасова // Экономический анализ: теория и практика, №6, 2006.- С. 11-15.
10. Перова В.И. Имитационное моделирование в среде ITNINK: модель кредитования предприятий / В.И. Перова, А.В. Матюков // Материалы международной научно-методической конференции “Болонский процесс: сотрудничество российских и европейских университетов”, Нижний Новгород: Изд-во “Пламя”, 2006.- С. 65-69.
11. Шебеко Ю. А. ITNINK - финансовым менеджерам (аналитический пакет нового поколения помогает в решении проблем) // [www.tora-centre.ru](http://www.tora-centre.ru)
12. Горбунов А. Р. Пакет структурного моделирования ITNINK: инвестиционные проекты, реинжиниринг, стратегия. – М.: ТОРА-Центр. 1997. - 24 с.