

## Коротко об авторах этой книги



Мария Каменнова — Председатель Совета Директоров компании «Весть-МетаТехнология». Основное достижение — «раскрутка» на российском рынке консалтинга в области высоких технологий как вида реального бизнеса. В настоящее время управляет проектами по реорганизации деятельности, внедрению функционально-стоимостного анализа и внедрению систем класса workflow на предприятиях.



Александр Громов — директор компании «Весть-МетаТехнология». С 1993 г. занимается внедрением методологий системного анализа при реорганизации деятельности предприятий, автор российской методологии системного управления качеством, занимается кризис-менеджментом, разработкой и внедрением методологии реструктуризации предприятий и их подготовкой к внедрению систем управления.



Окончил МИФИ в 1975 г. После окончания института занимался научной работой и преподавал. Кандидат технических наук, доцент. Имеет около 50 научных работ, Лауреат премии Минвуза. С 1999 г. генеральный директор компании «Весть-МетаТехнология». Является одним из организаторов продвижения методологии и семейства программных продуктов ARIS на российский рынок. Активно работает в области развития бизнеса по вопросам анализа, моделирования и оптимизации бизнес-процессов.



Антон Шматалюк — руководитель проектов компании «Весть-Метатехнология». С 1998 г. — сотрудник компании «Весть-МетаТехнология». Стоял у истоков освоения методологии и инструментария ARIS и продвижения их на российский рынок. В настоящий момент является руководителем проектов и ответственным за развитие технологий ARIS в применении их к задачам моделирования и анализа деятельности, внедрения информационных систем.

## Предисловие

Необходимость адаптации к изменениям, которые происходят в России как на макроэкономическом уровне, так и на уровне каждого российского предприятия, требует специальных методов и средств для «настройки» бизнес-процессов предприятий относительно быстро меняющихся внешних и внутренних условий и правил ведения бизнеса. Эти средства должны обеспечивать адекватную информацию о деятельности предприятия и предоставлять основу и методы для анализа этой информации в аспекте бизнес-процессов предприятия, их узких мест и возможностей оптимизации. Предлагаемая вниманию читателей книга М. С. Каменновой, А. И. Громова, М. М. Ферапонтова, А. Е. Шматалюка «Моделирование бизнеса. Методология ARIS» посвящена одному из наиболее продвинутых и полнофункциональных инструментов для моделирования и анализа бизнес-процессов и их окружения — системе ARIS 5.0, представляемой на мировом рынке немецкой компанией IDS Scheer AG.

Авторы книги стояли у истоков появления системы ARIS в России и ее осмысления. Все они — сотрудники компании «Вест-МетаТехнология», которая является российским партнером IDS Scheer AG с 1997 г. и почти 5 лет занимается работой, связанной с технической поддержкой продукта и его версий, локализацией, предоставлением услуг по настройке ARIS, переводом многочисленных методических и информационных материалов, разработкой учебных программ и собственной методологии использования ARIS в проектах по моделированию и анализу бизнес-процессов, их оптимизации и настройке, последующему внедрению информационных систем различного класса.

На сегодняшний день в России уже несколько тысяч зарегистрированных пользователей системы ARIS, а еще больше тех, у кого есть интерес к этому продукту и объективная потребность в его использовании.

Книга «Моделирование бизнеса. Методология ARIS» призвана в известной степени удовлетворить этот интерес и познакомить российских читателей с современными подходами и средствами для моделирования и анализа бизнес-систем.

Книга ориентирована на читателей с различным уровнем осведомленности о том, что такое моделирование и что такое ARIS. Те, кто уже пользуется системой и приобрел некоторые навыки работы с ней, найдут в книге систематизированную и детализированную информацию о принципах и методах работы в среде ARIS, ее основных функциональных модулях, методологических фильтрах, возможностях генерации отчетов, а также о многих известных и не очень известных приемах описания и моделирования бизнес-процессов. Для более опытных пользователей она может служить расширенным справочником по работе с ARIS.

Эту книгу настоятельно можно рекомендовать и тем, кто только начинает знакомство с методами и инструментами моделирования, системой ARIS и хотел бы понять, что дает эта система, как она устроена и как работает. Читателя, не очень искушенного в области системного анализа и моделирования, книга знакомит с такими понятиями, как система, бизнес-система, бизнес-процесс, модель бизнес-процесса, методы анализа бизнес-процессов, и лишь затем погружает в предметную область самой инструментальной системы ARIS.

Первые две главы книги представляют собой краткое введение в методы системного анализа и их применение для исследования деятельности организации. В них обсуждаются понятия структурного анализа и моделирования систем.

Глава 3 знакомит читателей с методологией ARIS, ее архитектурой, основными компонентами, типами моделей, объектов и связей.

## Предисловие

В главе 4 «крупными мазками» описаны основные принципы реализации инструментальной системы ARIS, а в главе 5 подробно рассматриваются все ее модули.

Последующие две главы содержат обзор моделей, входящих в состав методологических фильтров системы ARIS. В главе 6 описываются модели, включенные в простой методологический фильтр, а в главе 7 — модели, вошедшие в стандартный и расширенный методологические фильтры. Глава 8 дополняет предыдущее изложение описанием наиболее часто используемых моделей из полного методологического фильтра.

В главе 9 обсуждаются основные принципы моделирования в ARIS — как выбирать методологию и типы моделей в зависимости от задачи, до какого уровня детализации необходимо рассматривать модели, как проводить их анализ и т.д.

Глава 10 знакомит с принципами документирования моделей в системе ARIS. Это очень серьезный аспект, так как в конечном итоге все результаты работы в системе ARIS необходимо адекватно представлять тем, кто принимает стратегические и тактические решения при построении бизнеса и определении приоритета тех или иных задач.

В главе 11 приведен глоссарий ARIS.

При подготовке издания авторы, разумеется, не могли обойтись без помощи наших коллег. Особую благодарность хотелось бы выразить В. В. Крохину за его кропотливую работу по отбору и включению в книгу материалов, написанных авторами в разное время. Авторы также благодарны Олегу Зерняеву за предоставленные материалы по учебным курсам ARIS.

Искренние слова благодарности хотелось бы сказать в адрес наших коллег, которые подготовили книгу к печати — литературному редактору Е. В. Крестьяниновой и Н. Каменнову, создавшему оригинал-макет книги.

Жизнь не стоит на месте, и пока готовилось это издание, появились новые версии продукта, новые возможности. Коллектив авторов уже приступил к работе над следующей книгой, посвященной специализированным методам и модулям в рамках системы ARIS. В ней будут представлены:

- методология стоимостного анализа бизнес-процессов и модуль ARIS ABC;
- модуль для имитационного моделирования ARIS Simulation и методология работы с ним;
- специализированные средства ARIS for R/3, позволяющие с помощью ARIS выполнять соответствующие настройки в референтных (эталонных) моделях SAP R/3 и транслировать их из ARIS for R/3 непосредственно в SAP R/3;
- методология и программные средства для внедрения систем менеджмента качества (ИСО 9000-2000);
- методология и программные средства для измерения и анализа производительности бизнес-процессов;
- модули-интерфейсы, связывающие ARIS с различными системами других типов (CASE-системами, системами класса workflow).

Мы планируем выход этой книги в свет в начале 2002 г.

Мария Каменнова, кандидат  
технических наук Москва,  
июнь 2001 г.

## Содержание

Коротко об авторах этой книги .....	1
Предисловие.....	2
Содержание .....	4
<b>1. СИСТЕМНЫЙ АНАЛИЗ ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ .....</b>	<b>7</b>
1.1. Понятие организации.....	7
1.2. Функционально-ориентированная (иерархическая) организация .....	8
1.3. Процессно-ориентированная организация .....	9
1.4. Понятие системы .....	11
1.5. Системный подход .....	13
<b>2. СТРУКТУРНЫЙ АНАЛИЗ ОРГАНИЗАЦИИ .....</b>	<b>15</b>
2.1. Структура системы .....	15
2.2. Структурный анализ .....	16
2.3. Методология структурного анализа .....	18
2.4. Понятия модели и моделирования.....	20
2.5. Этапы структурного анализа .....	23
<b>3. МЕТОДОЛОГИЯ ARIS.....</b>	<b>26</b>
3.1. Общие сведения. Постановка задачи .....	26
3.2. Предназначение системы ARIS. Примеры успешного использования .....	26
3.3. Архитектура ARIS .....	28
3.4. Фазовая модель ARIS .....	31
3.5. Модели ARIS.....	33
3.6. Объекты моделей.....	37
3.7. Связи объектов .....	39
<b>4. ОСНОВНЫЕ ЭЛЕМЕНТЫ РЕАЛИЗАЦИИ СИСТЕМЫ ARIS .....</b>	<b>41</b>
4.1. Организация управления элементами ARIS .....	41
4.2. Атрибуты элементов ARIS .....	49
4.3. Методологические фильтры.....	53
<b>5. ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ARIS.....</b>	<b>55</b>
5.1. Обзор основных модулей .....	55
5.2. Проводник — ARIS EXPLORER .....	59
5.2.1. Основные элементы управления .....	59
5.2.2. Компоненты сетевого окружения ARIS — ARIS Network.....	64
5.3. Конструктор моделей — ARIS DESIGNER .....	69
5.3.1. Основные элементы управления .....	69
5.3.2. Моделирование .....	72
5.4. Атрибуты элементов ARIS — ARIS ATTRIBUTES.....	79
5.5. Проверка моделей — ARIS SEMANTIC CHECK.....	83
5.6. Отчеты о моделях — ARIS REPORT .....	83
5.7. Анализ и анимация моделей — APIS ANALYSIS .....	85
5.8. Управление изменениями — APIS CHANGE MANAGEMENT .....	87
5.9. Имитационное моделирование — APIS SIMULATION .....	90
5.10. Построение диаграмм и графиков — ARIS CHART .....	91
5.11. Администратор APIS — APIS ADMINISTRATOR.....	98
5.12. Идентификация элементов ARIS — APIS IDENTIFICATION.....	100
5.13. Конфигурирование системы — ARIS CONFIGURATION.....	104

5.14. КОНСОЛИДАЦИЯ ОБЪЕКТОВ БАЗ ДАННЫХ — ARIS CONSOLIDATION .....	105
5.15. ОБЪЕДИНЕНИЕ МОДЕЛЕЙ — APIS MERGE .....	108
5.16. ГЕНЕРАТОР МОДЕЛЕЙ — ARIS MODEL GENERATOR .....	109
5.17. ВАРИАНТЫ МОДЕЛЕЙ — ARIS VARIANTS .....	110
5.18. ЭКСПОРТ И ИМПОРТ ЭЛЕМЕНТОВ — ARIS EXPORT/IMPORT .....	114
5.19. ТЕКСТОВЫЙ РЕДАКТОР — ARIS RTF EDITOR .....	118
5.20. РЕДАКТОР СКРИПТОВ — ARIS SCRIPT EDITOR .....	118
5.21. ФУНКЦИОНАЛЬНО-СТОИМОСТНОЙ АНАЛИЗ — APIS ABC .....	119
5.22. ИНТЕГРАЦИЯ С ПРИКЛАДНЫМИ ИНФОРМАЦИОННЫМИ СИСТЕМАМИ — ARIS TOOL INTEGRATION .....	120
5.23. ПРЕОБРАЗОВАНИЕ БАЗ ДАННЫХ ARIS 3.x В БАЗЫ ДАННЫХ ARIS 5.0 — ARIS DATABASE CONVERTER .....	120
5.24. ARIS И ИНТЕРНЕТ — ARIS WEB PUBLISHER .....	120
<b>6. ПРОСТОЙ МЕТОДОЛОГИЧЕСКИЙ ФИЛЬТР. ОБЗОР МОДЕЛЕЙ.....</b>	<b>122</b>
6.1. ОРГАНИЗАЦИОННЫЕ МОДЕЛИ .....	122
6.1.1. <i>Организационная схема— Organizational chat</i> .....	122
6.2. ФУНКЦИОНАЛЬНЫЕ МОДЕЛИ .....	127
6.2.1. <i>Дерево функций — Function Tree</i> .....	127
6.3. МОДЕЛИ ДАННЫХ .....	131
6.3.1. <i>Модель технических терминов — Technical Term Models</i> .....	131
6.4. МОДЕЛИ ПРОЦЕССОВ/ УПРАВЛЕНИЯ.....	133
6.4.1. <i>Событийная цепочка процесса — Extended event driven process chain (eEPC)</i> .....	133
6.4.2. <i>Диаграмма окружения функции — Function allocation diagram</i> .....	143
6.4.3. <i>Производственный и офисный процессы — Industrial and Office process</i> .....	148
6.4.4. <i>Диаграмма цепочек добавленного качества — Value-added chain diagram (VAD)</i> .....	156
<b>7. СТАНДАРТНЫЙ И РАСШИРЕННЫЙ МЕТОДОЛОГИЧЕСКИЕ ФИЛЬТРЫ. ОБЗОР МОДЕЛЕЙ..</b>	<b>159</b>
7.1. ФУНКЦИОНАЛЬНЫЕ МОДЕЛИ .....	159
7.1.1. <i>Диаграмма целей — Objective diagram</i> .....	159
7.1.2. <i>Диаграмма типа прикладной системы — Application system type diagram (ASTD)</i> .....	161
7.2. МОДЕЛЬ ДАННЫХ .....	165
7.2.1. <i>Расширенная модель «сущность - отношение» — Extended entity -relationship model (eERM)</i> .....	165
7.2.2. <i>Диаграмма атрибутов eERM-модели — eERM Attribute allocation diagram</i> .....	172
7.2.3. <i>Диаграмма структуры знаний — Knowledge structure diagram</i> .....	174
7.3. МОДЕЛИ ПРОЦЕССОВ/ УПРАВЛЕНИЯ.....	179
7.3.1. <i>Диаграмма информационных потоков — Information flow diagram</i> .....	179
7.3.2. <i>Матрица выбора процессов — Process selection matrix</i> .....	181
7.3.3. <i>Диаграмма eEPC (в виде столбцов) — eEPC (column display)</i> .....	183
7.3.4. <i>Карта знаний — Knowledge map</i> .....	185
7.3.5. <i>Диаграмма цепочки процесса — Process Chain Diagram (PCD)</i> .....	189
7.3.6. <i>Диаграмма движения продуктов/услуг — Product/ Service exchange diagram</i> .....	191
7.3.7. <i>Дерево продуктов/услуг — Product/Service tree</i> .....	195
7.3.8. <i>UML-диаграмма действий — UML Activity diagram</i> .....	198
7.3.9. <i>UML-диаграмма класса — UML Class diagram</i> .....	202
7.3.10. <i>UML-диаграмма описания класса UML Class description diagram —</i> .....	205
7.3.11. <i>UML-диаграмма взаимодействия — UML Collaboration diagram</i> .....	206
7.3.12. <i>UML-диаграмма компонент — UML Component diagram</i> .....	208
7.3.13. <i>UML-диаграмма состояний — UML State chart diagram</i> .....	209
7.3.14. <i>UML-диаграмма использования приложений — UML Use case diagram</i> .....	210
<b>8. ОБЗОР НЕКОТОРЫХ МОДЕЛЕЙ, ВОШЕДШИХ В ПОЛНЫЙ МЕТОДОЛОГИЧЕСКИЙ ФИЛЬТР</b>	<b>213</b>
8.1. ДЕРЕВО ПРОДУКТОВ — PRODUCT TREE .....	213
8.2. МАТРИЦА ВЫБОРА ПРОДУКТА — PRODUCT SELECTION MATRIX .....	214
8.3. ДИАГРАММА ОКРУЖЕНИЯ ПРОДУКТА — PRODUCT ALLOCATION DIAGRAM.....	215
8.4. ДИАГРАММА ВЗАИМОДЕЙСТВИЙ — COMMUNICATION DIAGRAM.....	217
8.5. МОДЕЛЬ КОНКУРЕНЦИИ — COMPETITION MODEL .....	218

8.6. МОДЕЛЬ «ТЕХНИЧЕСКИЕ РЕСУРСЫ» — TECHNICAL RESOURCES .....	221
8.7 МОДЕЛИ, ПРЕДНАЗНАЧЕННЫЕ ДЛЯ ОПИСАНИЯ МАТЕРИАЛОВ .....	223
8.7.1. Диаграммы eEPC (с потоком материалов) — <i>Extended event driven process chain (material flow)</i> ....	223
8.7.2. Диаграмма материалов — <i>Material Diagram</i> .....	225
8.7.3. Диаграмма материальных потоков — <i>Material Flow Diagram</i> .....	226
8.8. ДИАГРАММА УПРАВЛЕНИЯ БИЗНЕСОМ — BUSINESS CONTROL DIAGRAM .....	227
8.9. МОДЕЛЬ СТРУКТУРЫ — STRUCTURING MODEL .....	228
8.10. ДИАГРАММА РОЛЕЙ — ROLE DIAGRAM .....	229
<b>9. ОБЩИЕ ПРИНЦИПЫ МОДЕЛИРОВАНИЯ В ARIS .....</b>	<b>230</b>
9.1. ВЫБОР ТИПОВ МОДЕЛЕЙ .....	230
9.2. СТРУКТУРИРОВАНИЕ ИНФОРМАЦИИ ПРИ МОДЕЛИРОВАНИИ .....	233
9.3. ВАРИАНТЫ МОДЕЛЕЙ .....	241
9.4. ОТНОШЕНИЯ МЕЖДУ МОДЕЛЯМИ .....	241
9.5. ОПРЕДЕЛЕНИЕ ГЛУБИНЫ МОДЕЛИРОВАНИЯ .....	242
9.6. ПРАВИЛА ИМЕНОВАНИЯ И ОФОРМЛЕНИЯ МОДЕЛЕЙ .....	245
9.6.1. Задание правил именования .....	245
9.6.2. Задание правил графического оформления моделей .....	249
9.7. ПРОВЕРКА МОДЕЛЕЙ .....	251
9.8. АНАЛИЗ МОДЕЛЕЙ .....	257
9.9. ОПРЕДЕЛЕНИЕ ПРАВ ДОСТУПА .....	260
9.9.1. Виды пользователей и их права .....	260
9.9.2. Использование паролей .....	264
9.9.3. Необходимость управления доступом .....	265
<b>10. ДОКУМЕНТИРОВАНИЕ МОДЕЛЕЙ .....</b>	<b>266</b>
10.1. ОБЩИЕ ПРИНЦИПЫ ДОКУМЕНТИРОВАНИЯ .....	266
10.2. ГЕНЕРАЦИЯ ОТЧЕТА ПО ЗАДАННОМУ СКРИПТУ .....	267
10.3. СТАНДАРТНЫЕ СКРИПТЫ ARIS .....	270
10.3.1. Общие сведения о файлах скриптов .....	270
10.3.3. Описание скриптов для отчетов по папкам базы данных .....	271
10.3.4. Описание скриптов для отчетов по моделям .....	273
10.3.5. Описание скриптов для отчетов по таблицам функционально-стоимостного анализа .....	276
10.3.6. Описание скриптов для отчетов по объектам .....	276
10.3.7. Описание скриптов для отчетов по фильтрам .....	278
10.4. РЕКОМЕНДАЦИИ ПО ДОКУМЕНТИРОВАНИЮ МОДЕЛЕЙ .....	279
10.5. СРЕДСТВА ДЛЯ СОЗДАНИЯ ОТЧЕТОВ В ARIS .....	280
10.5.1. Создание скриптов при помощи <i>Script Wizard</i> .....	280
10.5.2. Создание скриптов в модуле <i>ARIS Script Editor</i> .....	287
<b>11. ГЛОССАРИЙ ARIS: МОДЕЛИ, ВХОДЯЩИЕ В МЕТОДОЛОГИЧЕСКИЕ ФИЛЬТРЫ, ТИПЫ ОБЪЕКТОВ И ТИПЫ СВЯЗЕЙ .....</b>	<b>290</b>
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>333</b>

# 1. Системный анализ деятельности организации

## 1.1. Понятие организации

Международный стандарт ИСО 9000:2000 определяет *организацию* как группу работников и необходимых средств с распределением ответственности, полномочий и взаимоотношений. Организация может быть корпоративной, государственной или частной. Можно дать и другое определение: организация —

это систематизированное, сознательное объединение действий людей, преследующих достижение конкретных целей.

Понятие «организация» раскрывает приведенная на рис. 1 модель технических терминов ARIS (Architecture of Integrated Information Systems — архитектура интегрированных информационных систем, см. раздел 3.3).

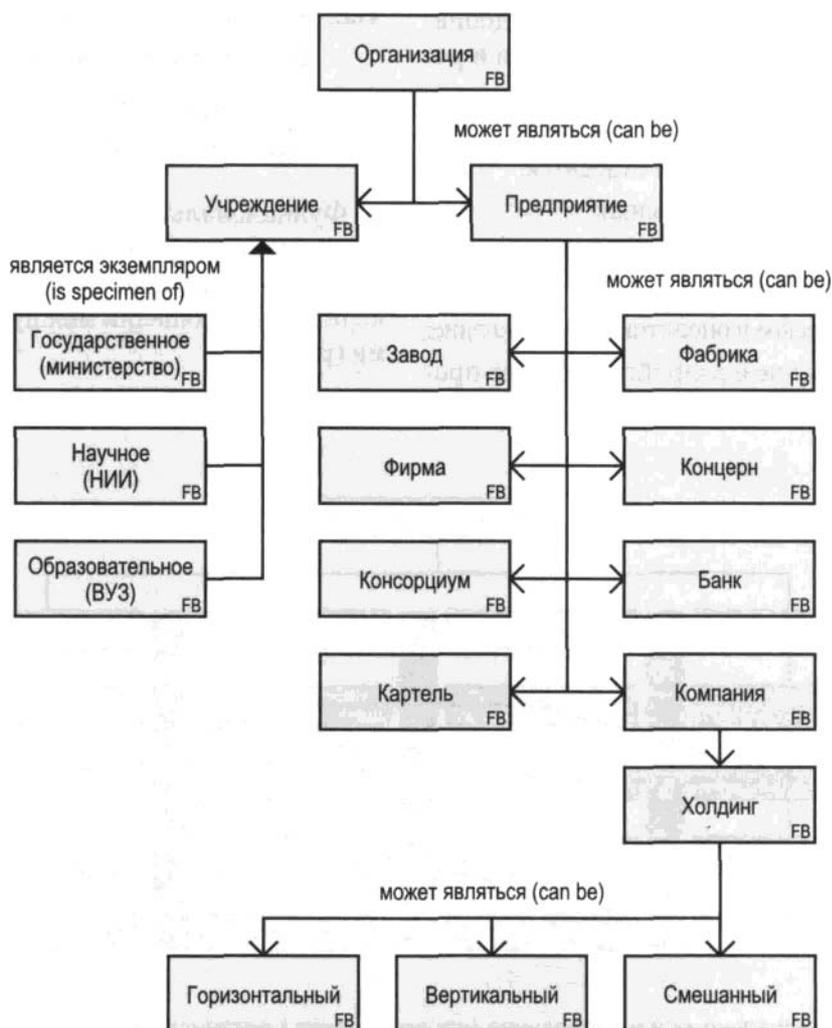


Рис. 1. Виды организаций, представленные с помощью модели технических терминов ARIS

## Моделирование бизнеса. Методология ARIS

В дальнейшем при изложении материала будет использоваться термин «организация». Каждая организация имеет несколько категорий заинтересованных сторон, имеющих свои нужды и ожидания. С точки зрения управления главными заинтересованными сторонами являются:

- заказчики и конечные пользователи;
- сотрудники организации;
- собственники и/или инвесторы;
- поставщики и партнеры;
- общество, интересы которого

представляют органы местного управления, и население, оказывающее влияние на организацию.

Каждая заинтересованная сторона надеется на свою собственную выгоду от той добавленной стоимости, которая появляется в результате деятельности организации.

Любая организация — многофункциональна. К ее основным функциям относятся:

- маркетинг и анализ рынка;
- стратегическое планирование деятельности предприятия;
- стратегическое и оперативное управление;
- планирование и разработка бизнес-процессов;

- проектирование и разработка продукции;
- производство продукции;
- поставка продукции;
- закупки материалов и комплектующих;
- техническое обслуживание и ремонт оборудования и прочие функции;
- оформление финансовых документов;
- подготовка кадров и управление персоналом.

Для выполнения этих функций организация должна быть соответствующим образом структурирована.

## 1.2. Функционально-ориентированная (иерархическая) организация

Функционально-ориентированные организации остаются неизменными и характеризуются вертикальной топологией структуры и иерархией отношений между подразделениями (рис. 2).

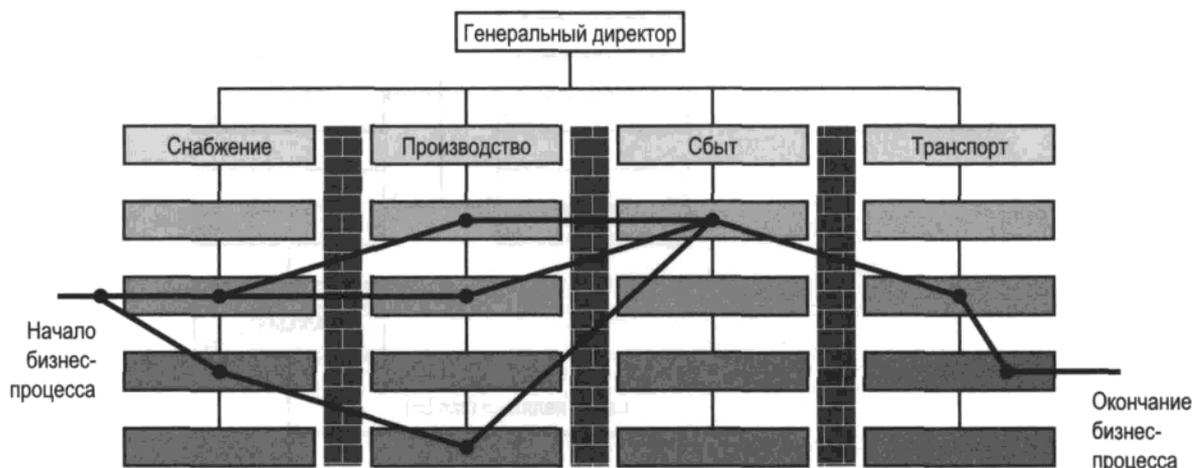


Рис. 2. Функционально-ориентированная (иерархическая) организация

### Системный анализ деятельности организации

В организации, имеющей функционально-ориентированную структуру, одно функциональное подразделение (закупки, производство, финансы и бухгалтерия) несет ответственность за все продукты и территории. Преимущество узкой специализации служащих «компенсируется» непомерными накладными расходами на коммуникации и координацию функциональных подразделений.

Функционально-ориентированные организации обладают рядом недостатков [ 1, 2], основными из которых являются:

- невозможность быстрой реакции на изменения;
- оторванность работающих от конечного результата;
- главным потребителем результатов труда работника является вышестоящий начальник;
- отсутствие ориентации на клиента;
- чрезвычайно усложнены взаимодействие и обмен информацией между подразделениями организации.

Альтернативой строго функциональной структуре является процессно-ориентированная структура.

### 1.3. Процессно-ориентированная организация

Понятие «процесс» — ключевое в современной теории управления бизнесом. Международный стандарт ИСО 9000:2000 определяет *процесс* как совокупность взаимосвязанных и взаимодействующих видов деятельности, преобразующих входы и выходы (рис. 3). Процесс включает одну или более связанных между собой процедур или функций, которые совместно реализуют некую задачу бизнеса — обычно в рамках организационной структуры. Он может выполняться в пределах одной организационной единицы, охватывать несколько единиц или даже несколько различных организаций, например, в системе «покупатель-поставщик».

Процесс обычно связан с операционными отношениями, например, процесс разработки нового изделия или процесс продаж. Введем некоторые термины, взаимосвязь которых представлена на рис. 3.

Различают основные и вспомогательные процессы. *Основные процессы* — это те процессы, которые добавляют новое качество продукции. *Вспомогательные процессы* формируют инфраструктуру организации.

*Владелец процесса* — лицо (или группа лиц), отвечающее за процесс и имеющее полномочия изменять его с целью усовершенствования.

*Границы процесса* — граница входа и граница выхода. Граница входа предшествует первой операции процесса, граница выхода следует за его последней операцией.

*Интерфейс процесса* — механизм (организационный, информационный, технический), посредством которого процесс взаимодействует с предшествующим и последующим процессами.

В соответствии с новым взглядом на организацию работа должна быть организована вокруг процессов. По словам Хаммера и Чампи [3], «не товары, а процессы их создания приносят компаниям долгосрочный успех». Целью организации должно быть совершенствование бизнес-процессов для преодоления их фрагментарности и для достижения существенных улучшений в ключевых показателях результативности — затраты, качество, уровень обслуживания и оперативность.

Несмотря на явные преимущества процессно-ориентированного устройства компании, — добиться создания такой структуры в чистом виде не представляется возможным.

Первым шагом проекта по оптимизации деятельности должно стать выделение основных продуктов компании и выстраивание процессов в соответствии с продуктовыми линиями. Это позволяет получить продуктовые «срезы» бизнес-процессов, протекающих в организации.

### Моделирование бизнеса. Методология ARIS

Однако всегда существует несколько функциональных подразделений, которые принимают участие в обслуживании всех продуктовых линий, например, бухгалтерия, транспортный цех и т.д. Перестроить данные направления, разбив их на продуктовые срезы компании, крайне сложно, так как это повлечет за собой массу дополнительных проблем и расходов.

Таким образом, задача формализации и оптимизации деятельности сводится к выделению бизнес-процессов в соответствии с продуктовыми линиями и функциональными подразделениями, и увязке их в сквозные процессы компании, нацеленные в первую очередь на создание продуктов и предоставление услуг клиентам.

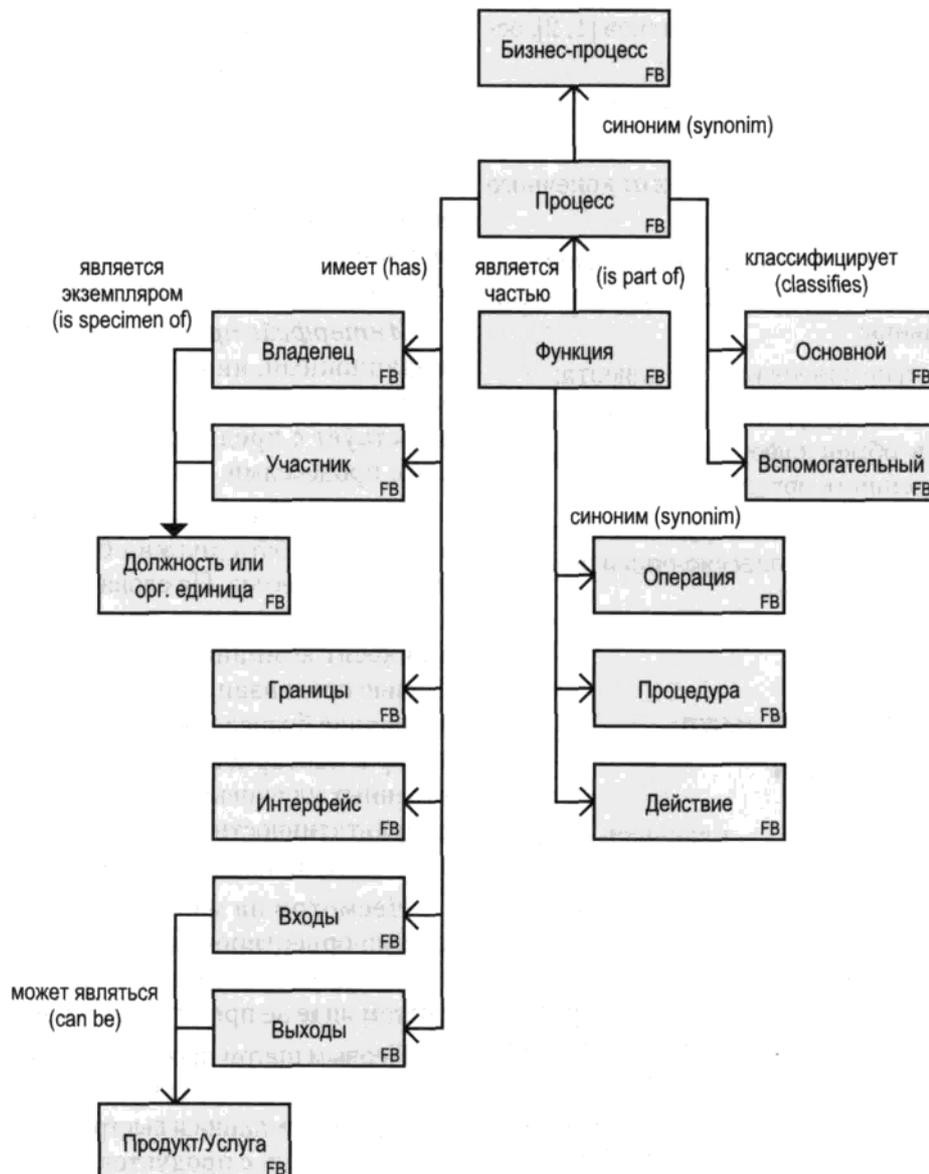


Рис. 3. Понятие «процесс», представленное с помощью модели технических терминов ARIS

## 1.4. Понятие системы

Любая организация является сложной социально-технической системой. Термин «система», употребляемый в современной практике, имеет множество значений и смысловых нюансов. Это приводит к необходимости выделить те значения, которые имеют непосредственное отношение к системному анализу деятельности организации. Далее приведены три определения, которые представляются наиболее удачными.

Первое из них дано в Международном стандарте ИСО 9000:2000 «Системы менеджмента качества. Основные положения и словарь». *Система* — это совокупность взаимосвязанных и взаимодействующих элементов. Следует отметить, что в современном менеджменте качества уделяется большое внимание системному подходу к деятельности организации.

Российский энциклопедический словарь трактует понятие «система» следующим образом: *система* (от греческого *Systema* — целое, составленное из частей) — множество элементов, находящихся в отношениях и связях друг с другом, образующих определенную целостность, единство.

И, наконец, третье определение, взятое из [4]: *система* — совокупность связанных между собой и с внешней средой элементов и частей, функционирование которых направлено на получение конкретного результата.

Как всякое фундаментальное понятие, термин «система» лучше всего конкретизируется при рассмотрении его основных свойств [5]. Для системы характерны следующие основные свойства:

- *целенаправленность* — определяет поведение системы;
- *сложность* — зависит от множества входящих в систему компонентов, их структурного взаимодействия, а также от сложности внутренних и внешних связей и их динамичности;
- *делимость* — система состоит из ряда подсистем или элементов, выделенных по определенному признаку, отвечающему конкретным целям и задачам;

- *целостность* — функционирование множества элементов системы подчинено единой цели. При этом система проявляет так называемые интегративные свойства, т.е. свойства, присущие системе в целом, но отсутствующие в отдельно взятых ее элементах;

- *многообразие элементов и различие их природы* — это связано с их функциональной специфичностью и автономностью;

- *структурированность* — определяется наличием установленных связей и отношений между элементами внутри системы, распределением элементов системы по уровням иерархии.

Исходной характеристикой системы является ее противопоставление окружению, или среде. *Среда* — это все то, что не входит в систему. Среда представляет собой совокупность всех систем, кроме исследуемой, выделенной, интересующей нас в настоящий момент части реального окружающего мира. Поэтому можно сказать, что система — это конечное множество объектов, каким-то образом выделенное из среды посредством границы системы. Понятие «границы» в целом ряде случаев весьма условно, и при моделировании необходимо четко определить, где кончается система, а где начинается среда.

Между средой и бизнес-системой, которой является организация (рис. 4), существует множество взаимных связей, с помощью которых реализуется процесс взаимодействия среды и системы.

По входной и выходной связям между системой и средой путем взаимной передачи происходит обмен материальными, финансовыми, энергетическими, информационными и иными элементами. Элементы, передаваемые системой во внешнюю среду, будем называть конечными продуктами деятельности системы, а передаваемые из среды в систему — ресурсами.

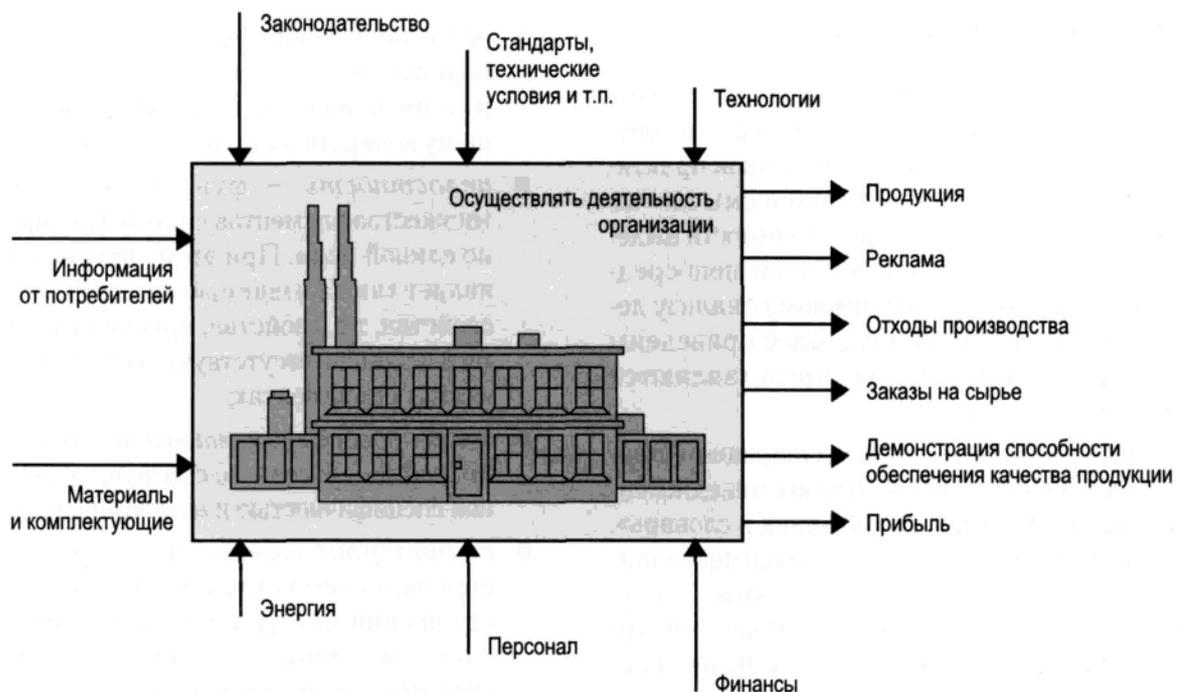


Рис. 4. Связи системы-организации с внешней средой

*Цель системы* — достижение и сохранение желаемого состояния или желаемого результата поведения системы. Применительно к организации более подходит определение цели, данное в [6]. Цель организации — стремление к максимальному результату, выражаемому в максимизации ценности капитала, при постоянном сохранении определенного уровня ликвидности и достижении целей производства и сбыта с учетом социальных задач. Вспомогательной стоимостной целью является стремление к оптимальной расчетной прибыли за период.

*Задача системы* — описание способа (технологии) достижения цели, содержащее указание на цель с желаемыми конкретными числовыми (в том числе временными) характеристиками.

*Система целей* — совокупность взаимоувязанных целей. В соответствии с определением понятия «система» для одного и того же объекта может быть рассмотрено несколько систем целей, т.е. использовано несколько оснований для их классификации, например:

- стратегические и тактические цели;
- долгосрочные (выполнение через несколько лет) и краткосрочные (выполнение через год и ранее) цели;
- производственные, финансовые, социальные цели, цели повышения качества продукции и т.п.

Древовидная система целей (рис. 5) включает как минимум глобальную цель — существование организации и две главные цели — цель функционирования (выпускать продукцию) и цель развития (развиваться).



Рис. 5. Цели организации, представленные в виде диаграммы целей ARIS

Таким образом, система представляет собой упорядоченное подмножество объектов, интенсивность взаимосвязей которых превышает интенсивность отношений с объектами, не входящими в данное подмножество, т.е. с внешней средой. *Объект (элемент, компонент)* — часть системы, выделенная по какому-либо признаку, сформулированному заинтересованным лицом. При этом объекты системы и отношения между ними выделяются в зависимости от точки зрения заинтересованного лица или группы лиц, например, одно и то же предприятие может рассматриваться как производственная, организационно-экономическая или социальная система. *Выбор точки зрения* — категория системного анализа, характеризующаяся выделением определенных аспектов рассмотрения проблемы и применением особой терминологии, соответствующей этим аспектам.

## 1.5. Системный подход

*Системный подход* — это методология специального научного познания и социальной практики, а также объяснительный принцип, в основе которого лежит исследование объектов как систем [5].

Методологическая специфика системного подхода определяется тем, что он ориентирует исследование на:

- раскрытие *целостности* объекта и обеспечивающих его механизмов;
- выявление многообразных типов *связей* сложного объекта;
- сведение этих связей в единую теоретическую картину.

Системный подход реализует представление сложного объекта в виде иерархической системы взаимосвязанных *моделей*, позволяющих фиксировать целостные свойства объекта, его структуру и динамику.

Методология ARIS, которой посвящена данная книга, основана на применении системного подхода в полной мере.

*Системный анализ* — совокупность методологических средств, используемых для подготовки и обоснования решений по сложным проблемам социального, технического и экономического характера. Он основывается на системном подходе, а также на ряде математических дисциплин и современных методов управления. Основной процедурой системного анализа является построение обобщенной модели, адекватно отображающей интересующие исследователя свойства реальной системы и ее взаимосвязи.

Главной задачей системного анализа является поиск путей по превращению сложного в простое, по разложению труднопонимаемой задачи на ряд задач, имеющих решение.

Принципы системного анализа [5]:

- *Оптимальность.* В результате анализа необходимо найти оптимальное решение задачи.

- *Эмерджентность.* Этот принцип предполагает следующее важное свойство системы: чем больше система и чем больше различие между частью и целым, тем выше вероятность того, что свойства целого могут сильно отличаться от свойств его частей. Принцип эмерджентности позволяет выявить несовпадение локальных оптимумов целей системы с глобальным оптимумом системы. Этот фактор необходимо учитывать при проведении системного анализа деятельности организации, поскольку он играет важную роль.

- *Системность.* Исследование объекта, с одной стороны, как единого целого, а с другой, как части более крупной системы, с которой объект находится в определенных отношениях.

- *Иерархичность.* Определение в системе структурных отношений, характеризующих упорядоченностью, организованностью взаимодействий между отдельными ее уровнями по вертикали. Большинство организаций представляют собой сложные системы, и необходимость иерархического построения этих систем обусловлена тем, что управление в них связано с переработкой и использованием больших объемов информации.

- *Интеграция.* Изучение интеграционных свойств и закономерностей системы.

- *Формализация.* Получение комплексных количественных характеристик.

Системный анализ — наиболее эффективный метод, применяемый при построении модели организации. Однако при проведении его могут возникать проблемы, обусловленные

сложностью социально-технических систем, так и проблемы политического характера, если заинтересованные группы конфликтуют. При сочетании этих проблем системный анализ становится трудным и кропотливым делом, при котором аналитик должен играть роль посредника между заказчиком и исполнителем.

Руководители предприятий обычно интуитивно понимают свои проблемы, но не могут объяснить их, и, кроме того, они часто имеют весьма туманное представление о том, какую пользу могут принести им информационные технологии. Разработчики же с энтузиазмом говорят о существующих возможностях в области построения систем обработки данных, но они, как правило, не имеют информации о том, что именно является наилучшим для той или иной организации.

Основополагающая концепция состоит в построении при помощи графических методов системного анализа совокупности моделей различных аспектов деятельности организации, которые дают возможность управленцам и аналитикам получить ясную общую картину бизнес-процессов.

Системный подход применяется для решения различного рода сложных задач, в число которых входят:

- совершенствование системы управления организации и анализ ее деятельности;
- подготовка к внедрению системы управления предприятием;
- внедрение систем менеджмента качества и их сертификация;
- оптимизация, инжиниринг и реинжиниринг бизнес-процессов;
- внедрение информационных систем на предприятии;
- документирование корпоративных знаний, в том числе в виде моделей прототипов.

## 2. Структурный анализ организации

### 2.1. Структура системы

*Структура системы* — это совокупность устойчивых связей объекта, обеспечивающих его целостность и тождественность самому себе, т.е. сохранение основных свойств при различных внешних и внутренних изменениях. С другой стороны, структура системы — частичное упорядочение элементов системы и отношений между ними по какому-либо признаку. Структура невозможна вне системы, равно как и система всегда структурирована.

Структуризация направлена на:

- выявление реальных целей системы;
- выяснение альтернативных путей достижения этих целей;

- достижение взаимосвязей между элементами;
- получение возможности моделирования системы.

Переход от системы к структуре может быть осуществлен только при условии, что найдены элементы и их устойчивые отношения. Причем, как правило, существует большое число критериев, по которым выбираются составляющие систему элементы. Таким образом, можно говорить о множественности структур системы.

В организациях может быть выделено несколько типовых структур (рис. 6).

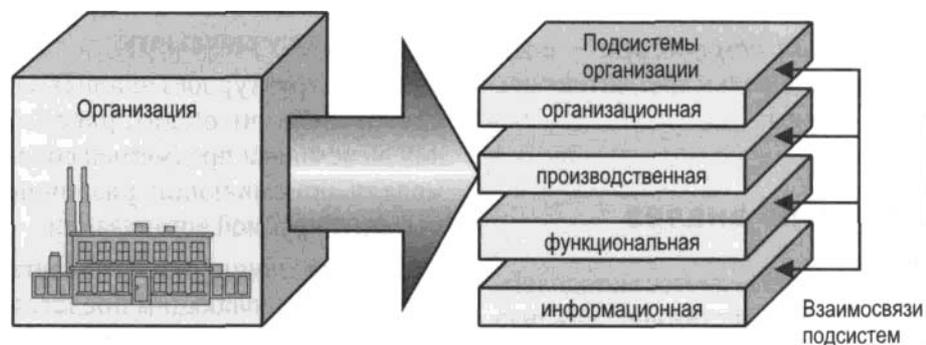


Рис. 6. Разбиение организации на структурные подсистемы

Введем несколько определений.

*Организационная структура* — это структура, элементами которой являются подразделения организации разного уровня иерархии, а отношениями — отношения входимости и руководства-подчинения.

*Производственная структура* — часть организации, выполняющая задачи оперативного управления производством и обеспечивающая выпуск продукции и/или предоставление услуг.

*Функциональная структура* — структура, элементами которой являются функции, реализуемые подразделениями предприятия, а

отношениями — связи, обеспечивающие передачу между элементами предметов труда.

*Информационная структура* — совокупность центров производства, сбора, анализа и распространения информационных потоков.

### Моделирование бизнеса. Методология ARIS

*Структура выходов организации* — совокупность материальной и нематериальной продукции, являющейся результатом деятельности организации и поставляемой ею во внешнюю (по отношению к ней) среду.

*Структура входов организации* — совокупность материальной и нематериальной продукции, используемой для осуществления деятельности организации.

*Юридическая структура* — совокупность бизнес-единиц с множеством организационных, административно-правовых отношений между ними, а также отношений собственности и контроля.

*Финансово-экономическая (финансовая) структура* — совокупность центров учета с финансовыми потоками между ними.

*Штатная структура* — состав подразделений и перечень должностей, размеры должностных окладов и фонд заработной платы.

*Социальная структура* — разбиение персонала организации на группы по социальным показателям.

*Территориальная структура* — совокупность мест расположения элементов организационной структуры.

## 2.2. Структурный анализ

Структурный анализ является методологической разновидностью системного анализа. Он был разработан в 60-70-х годах XX века Дугласом Т. Россом в виде методологии SADT (Structured Analysis and Design Technique)— технология структурного анализа и проектирования [7].

В основе структурного анализа лежит выявление структуры как относительно устойчивой совокупности отношений, признание методологического примата отношений над элементами в системе, частичное отвлечение от развития объектов. Основным понятием структурного анализа служит *структурный элемент (объект)* — элемент, выполняющий одну из элементарных функций, связанных с моделируемым предметом, процессом или явлением.

Структурный анализ предполагает исследование системы с помощью ее графического модельного представления,

которое начинается с общего обзора и затем детализируется, приобретая иерархическую структуру со все большим числом уровней. Для такого подхода характерны:

- разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 9);
- ограниченный контекст, включающий лишь существенные на каждом уровне детали;
- использование строгих формальных правил записи;
- последовательное приближение к конечному результату.

*Цель* структурного анализа заключается в преобразовании общих, расплывчатых знаний об исходной предметной области в точные модели, описывающие различные подсистемы моделируемой организации.

*Декомпозиция* (рис. 7) является условным приемом, позволяющим представить систему в виде, удобном для восприятия, и оценить ее сложность. В результате декомпозиции подсистемы по определенным признакам выделяются отдельные структурные элементы и связи между ними. Декомпозиция служит средством, позволяющим избежать затруднений в понимании системы. Глубина декомпозиции определяется сложностью и размерностью системы, а также целями моделирования.

## Структурный анализ организации

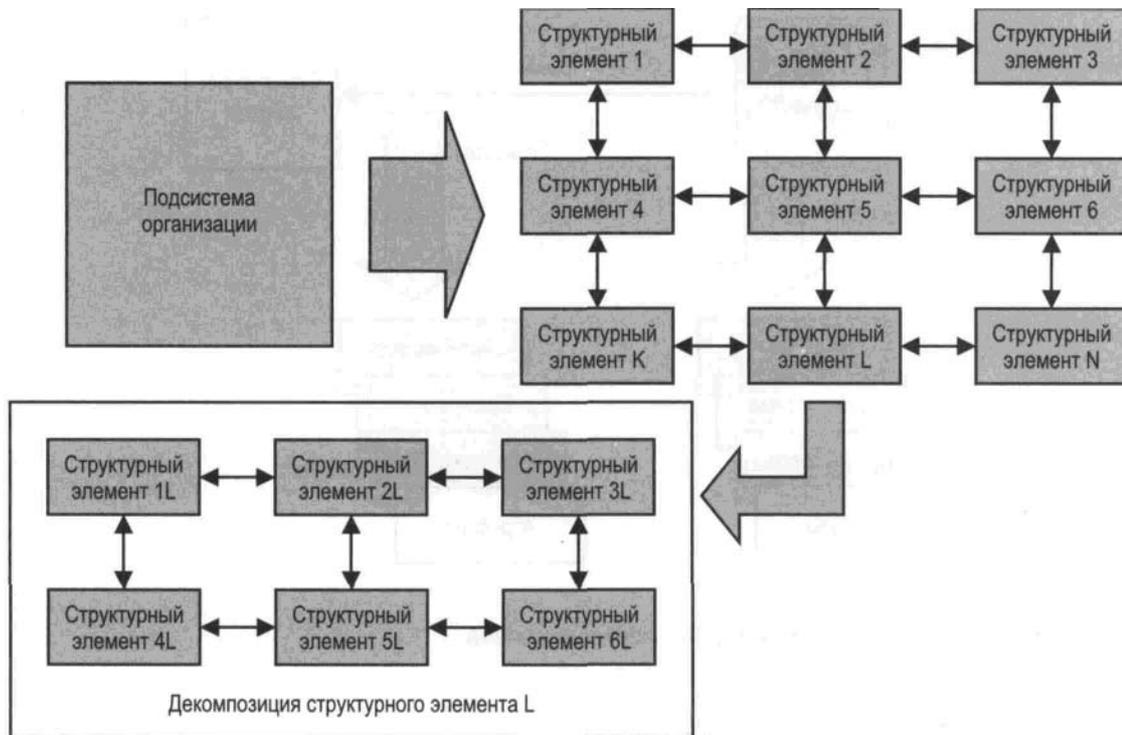


Рис. 7. Декомпозиция подсистемы организации на структурные элементы

Методология ARIS также использует декомпозицию и позволяет детализировать предмет моделирования с помощью альтернативных или дополняющих друг друга моделей.

Следует помнить, что ни одна отдельно взятая подсистема не может обеспечить моделирование бизнес-процессов полностью. Поэтому для получения целостной картины деятельности организации необходимо взять за основу описание одной из выделенных структур и интегрировать его с остальными. Как показывает практика, основой для такой интеграции чаще всего служит функциональная или информационная подсистема.

Любая организация, как правило, имеет большое количество подсистем, поэтому число структурных элементов и связей между ними весьма велико.

Каждый структурный элемент (или объект) и связь обладают определенными свойствами, которые должны быть описаны (рис. 8). Одной из разновидностей свойств являются атрибуты. *Атрибут* — необходимое, существенное, неотъемлемое свойство объекта. Естественно, что разные структурные элементы имеют разные атрибуты.

Каждый объект или связь имеет также набор характеристик (рис. 8), при помощи которых можно задать количественные и качественные характеристики моделируемых элементов. В частности, для каждой функции можно задать ее имя, уникальный код в проекте, автора, время и дату создания, детальное описание, пример реализации, временные и стоимостные затраты на выполнение данной функции и т. д. Все указанные характеристики объектов и связей формализованы и используются при проведении анализа или составлении отчета.

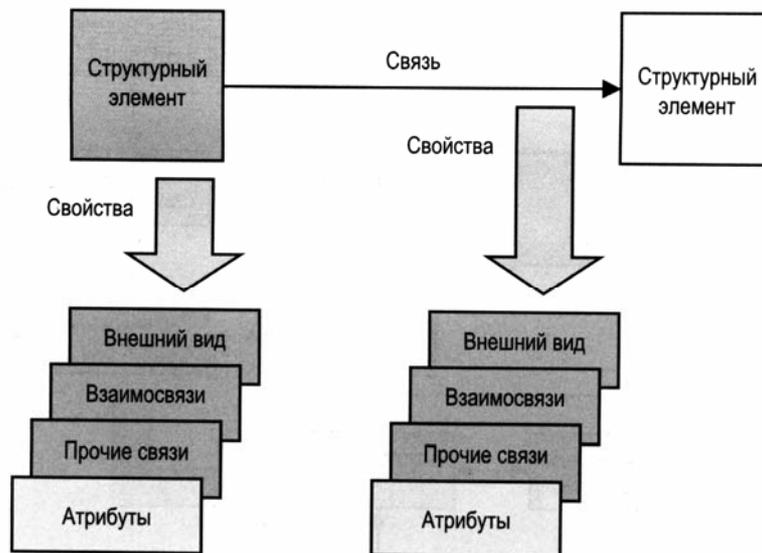


Рис. 8. Характеристики структурных элементов и связей

### 2.3. Методология структурного анализа

Структурный анализ как совокупность методов моделирования сложных систем вследствие большой размерности решаемых задач должен опираться на мощные средства компьютерной поддержки, обеспечивающей автоматизацию труда системных аналитиков. Такими средствами являются CASE-системы (Computer Aided Software Engineering).

Архитектура большинства CASE-систем основана на парадигме «методология — модель — нотация — средства» (рис. 9).

*Методология* структурного анализа представляет методы и средства для исследования структуры и деятельности организации. Она определяет основные принципы и приемы использования моделей.

*Модель* — это совокупность символов (математических, графических и т.п.), которая адекватно описывает некоторые свойства моделируемого объекта и отношения между ними.

*Нотации* — система условных обозначений, принятая в конкретной модели.

*Средства* — аппаратное и программное обеспечение, реализующее выбранную методологию, в том числе построение соответствующих моделей с принятой для них нотацией.

При моделировании систем вообще и, в частности, для целей структурного анализа используются различные модели, отображающие:

- функции, которые система должна выполнять;
- процессы, обеспечивающие выполнение указанных функций;
- данные, необходимые при выполнении функций, и отношения между этими данными;
- организационные структуры, обеспечивающие выполнение функций;
- материальные и информационные потоки, возникающие в ходе выполнения функций.

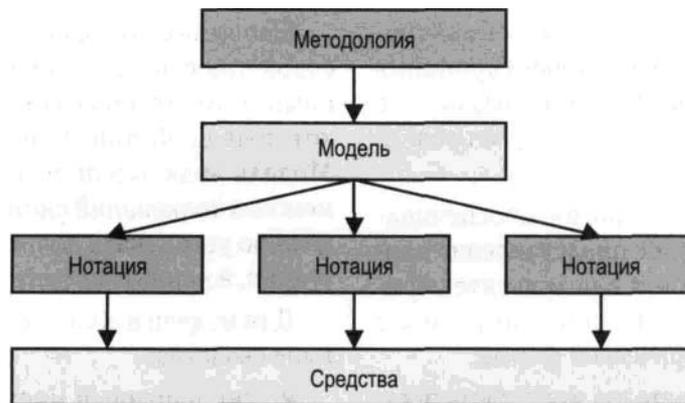


Рис. 9. Архитектура CASE-систем

Среди многообразия средств, предусмотренных для проведения структурного анализа, наиболее часто и эффективно применяются:

- DFD(Data Flow Diagrams)—диаграммы потоков данных в нотациях Гейна-Сарсона, Йордона-Де Марко и других, обеспечивающие требования анализа и функционального проектирования информационных систем;
- STD (State Transition Diagrams) — диаграммы перехода состояний, основанные на расширениях Хартли и Уорда-Меллора для проектирования систем реального времени;
- ERD (Entity-Relationship Diagrams) — диаграммы «сущность-связь» в нотациях Чена и Баркера;
- структурные карты Джексона и/или Константайна для проектирования межмодульных взаимодействий и внутренней структуры объектов;
- FDD (Functional Decomposition Diagrams) — диаграммы функциональной декомпозиции;
- SADT (Structured Analysis and Design Technique) — технология структурного анализа и проектирования;
- семейство IDEF (Integration Definition for Function Modeling):

IDEFO — методология функционального моделирования, являющаяся составной частью SADT и позволяющая описать бизнес-процесс в виде иерархической системы взаимосвязанных функций;

IDEF1 — методология анализа и изучения взаимосвязей между информационными потоками в рамках коммерческой деятельности предприятия;

IDEF1X — методология информационного моделирования, основанная на концепции «сущность-связь», предложенной Ченом [15]. Применяется для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы и обеспечивающий универсальное представление структуры данных в рамках предприятия, независимое от конечной реализации базы данных и аппаратной платформы;

IDEF3 — методология документирования технологических процессов, предприятия, позволяющая моделировать их сценарии посредством описания последовательности изменений свойств объекта в рамках рассматриваемого процесса;

### Моделирование бизнеса. Методология ARIS

- IDEF4 — методология объектно-ориентированного проектирования для поддержки проектов, связанных с объектно-ориентированными реализациями;

- IDEF5 — методология, обеспечивающая наглядное представление данных, полученных в результате обработки онтологических запросов, в простой, графической форме.

При помощи этих методов могут быть построены логические модели исходной и реорганизованной систем управления организацией.

## 2.4. Понятия модели и моделирования

Создаваемая модель должна давать ответ на следующие вопросы:

- Кто из сотрудников организации должен выполнять конкретные функции?

- При каких условиях нужно выполнять функцию?

- Что должен сделать сотрудник в рамках данной функции?

- Каким образом следует ее выполнять?

- Какие ресурсы при этом необходимы?

- Каковы результаты выполнения функции?

- Какие информационные средства нужны?

- Каким образом все это согласовать?

- Как все это можно осуществить наиболее эффективно?

- Как можно изменить или построить бизнес-процесс?

- Как снизить риск и повысить эффективность изменений?

Напомним, что модель представляет собой совокупность объектов и отношений между ними, которая адекватно описывает лишь некоторые свойства моделируемой системы. Модель является лишь одним из многих возможных толкований системы. Это толкование должно устраивать пользователя в данной ситуации, в данный момент времени.

Для модели в общем случае характерны четыре свойства:

- уменьшенный масштаб (размер) модели, точнее, ее сложность, степень которой всегда меньше, чем у оригинала. При построении модели сознательно вводятся упрощения;

- сохранение ключевых соотношений между разными частями;

- работоспособность, т.е. возможность в принципе работать, как оригинал-моделируемый объект (во всяком случае, похожим образом);

- адекватность действительным свойствам оригинала (степень достоверности).

Важно также подчеркнуть, что любая модель отражает точку зрения той или иной группы проектировщиков. Каждой модели присущи свои цели и задачи, и поэтому объект бизнеса, представляющий собой сложный комплексный организм, как правило, описывается некоторым набором моделей, в совокупности образующих общую модель данной бизнес-системы.

Использование множества моделей приводит к необходимости их классифицировать. Обоснованная классификация объектов представляет собой их условное группирование по заданным признакам в соответствии с определенной целью. При различных целях одни и те же объекты могут классифицироваться по-разному. Классификация не является самоцелью, она диктуется потребностями теории и практики. Целесообразная классификация моделей обеспечивает удобство при выборе методов моделирования и получение желаемых результатов.

### Структурный анализ организации

К важнейшим признакам, по которым проводится классификация моделей, относятся:

1) закон функционирования и характерные особенности выражения свойств и отношений оригинала;

2) основания для преобразования свойств и отношений модели в свойства и отношения оригинала [9].

По первому признаку модели делятся на логические, материальные и семантические, или вербальные (рис. 10). *Логические* модели функционируют по законам логики в сознании человека или в компьютере, работающем под управлением написанной человеком программы. *Материальные* модели функционируют в соответствии с объективными законами природы. *Семантические*, или *вербальные*, модели являются словесными описаниями объектов моделирования. Они применяются в ряде случаев, в частности на начальных этапах моделирования деятельности организации, при опросе – экспертами персонала с целью получения необходимой информации. Основная проблема, возникающая при построении вербальных моделей бизнес-процессов организации, заключается в установлении эффективного взаимодействия между экспертами предметной области (сотрудниками организации) и специалистами в области моделирования.

*Образные*, или *иконические*, модели выражают свойства оригинала с помощью наглядных образов, имеющих прообразы среди объектов материального мира. Набор моделей ARIS включает несколько моделей, которые по своей сути являются образными, или иконическими. Это, например, модели «Производственный процесс» (рис. 11), «Офисный процесс» и другие.

*Знаковые* (*символические*) модели выражают свойства моделируемой системы с помощью условных знаков или символов.

*Образно-знаковые* модели совмещают в себе признаки образных и знаковых моделей. Подавляющее большинство моделей ARIS являются образно-знаковыми.

*Функциональные*, *геометрические* и *функционально-геометрические* модели отражают соответственно только функциональные, только пространственные и одновременно функциональные и пространственные свойства оригинала. В методологии ARIS эти модели не используются.

По второму признаку модели делятся на условные, аналогичные и математические (рис. 10). *Условные* модели выражают свойства и отношения оригинала на основании принятого условия или соглашения. У таких моделей сходство с оригиналом может совершенно отсутствовать. Практически все модели ARIS являются условными. Следует отметить, что образные и образно-знаковые модели относятся тоже к условным.

*Аналогичные* модели обладают сходством с оригиналом, достаточным для перехода к оригиналу на основании умозаключения по аналогии. Такие модели также не используются в ARIS.

*Математические* модели обеспечивают переход к оригиналу, фиксацию и исследование его свойств и отношений с помощью математических методов. Математические модели обладают важными достоинствами — четкостью, возможностью строгой дедукции, проверяемостью. Однако в целом ряде случаев при построении математических моделей, например для описания процесса производства стали, могут возникнуть практически непреодолимые трудности. Тем не менее математические модели иногда используются в ARIS, в частности, при расчетах в ходе функционально-стоимостного анализа.

Можно провести квалификацию моделей в зависимости от их назначения. С точки зрения учета временного фактора выделяют статичные, имитационные и динамические модели.

*Статичные* модели описывают содержательную сторону системы, не изменяющуюся во времени. Они могут быть функционально-информационными, т.е. описывать структуру информации, на основе которой функционирует система, и структурными, т.е. описывать структуру системы.

### Структурный анализ организации

При моделировании организаций проводится главным образом условное моделирование, т.е. предполагается замещение оригинала условной моделью, представляющей его только в рамках договоренности о смысле, приписанном этой модели. В связи с этим вопрос о нотациях, используемых в знаковых и образно-знаковых моделях, приобретает большое значение.

К нотации модели предъявляются следующие основные требования:

простота — простое при прочих равных условиях предпочтительнее сложного;

наглядность — хотя бы отдаленное сходство с оригиналом облегчает использование модели;

индивидуальность — достаточное отличие от других обозначений;

однозначность — недопустимость обозначения одним символом различных объектов;

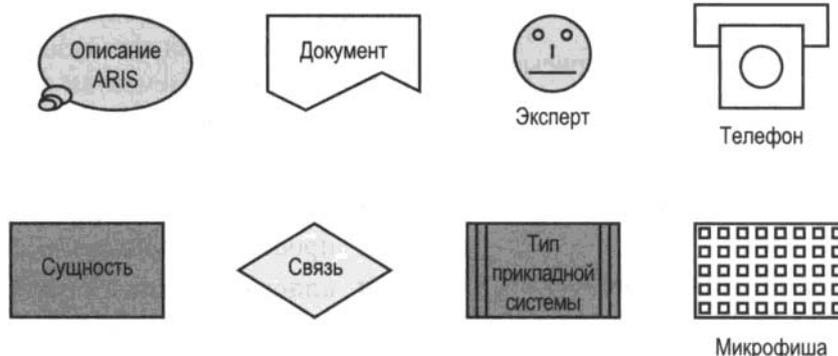


Рис. 12. Обозначение объектов в диаграмме структуры знаний ARIS

- единообразие — применение аналогичных правил при моделировании однородных объектов;
- определенность — четкие правила использования модели;
- учет устоявшихся традиций.

Нотация графической модели предполагает наличие:

- строго определенного набора взаимосвязанных графических изображений — элементов графического языка;
- различных типов связи между ними;
- фрагментов текста (естественного языка);
- встроенных объектов;
- глоссария.

Графический язык обеспечивает структуру и точную семантику естественному языку модели, организует естественный язык определенным и однозначным способом, что позволяет описывать весьма сложные модели.

Синтаксис графического языка содержит, как правило, разноцветные геометрические фигуры (прямоугольники, квадраты, параллелограммы, эллипсы, треугольники) и условные изображения разного рода (рис. 11-12).

Встроенные объекты — объекты других программных систем (Word, Excel, математические пакеты) — улучшают информационную насыщенность модели, делают ее более полной.

Глоссарий помогает пользователям разобраться с терминологией модели, облегчая тем самым ее понимание и использование.

## 2.5. Этапы структурного анализа

Проведение структурного анализа организации предполагает нескольких этапов:

- построение иерархии целей оптимизации деятельности организации;
- выбор методологии;
- выбор моделей;
- анализ деятельности организации;
- разработка моделей в соответствии с иерархией целей;
- оптимизация моделей;
- реорганизация деятельности.

На первом этапе выявляются и описываются цели, которые планируется достичь в ходе структурного анализа деятельности организации. Их, как правило, бывает несколько. В связи с этим цели необходимо ранжировать, выстроить их иерархию.

Когда цели реорганизации деятельности известны, появляется возможность для выбора методов проведения структурного анализа. Жестких алгоритмов выбора их не существует. Методология структурного анализа предполагает использование одной или нескольких моделей.

Определив цели анализа и выбрав инструменты для его проведения, необходимо детально изучить, как функционирует организация. Целью изучения является сбор данных для построения моделей, отображающих деятельность организации. Основными принципами проведения изучения деятельности организации являются:

- целенаправленность;
- комплексность;
- планомерность;
- организационно-методическая целостность.

Эти же принципы должны быть реализованы и в методике, включающей описание программы действий, изучаемых объектов, степени детализации изучения, методов сбора данных и правил их обработки. Такая методика обеспечивает стандартизацию изучения предметной области и формализованное представление данных.

Сбор информации производится в рамках всех основных структур организации. Большая часть собираемой информации не является очевидной, сформулированной и однозначной. В связи с этим перед началом моделирования необходимо выявить основные структурообразующие элементы системы управления анализируемой организации и зафиксировать их. К таким элементам относятся:

- организационная структура компании;
- структура территории;
- состав и структура основных бизнес-процессов компании;
- классификация и структура основных рабочих документов;
- классификация и структура информационных систем.

Организационная структура является наиболее очевидной составляющей любой компании. Однако и здесь могут быть проблемы. Так, проблема возникает при наличии прямой (дисциплинарной) подчиненности одного организационного элемента другому и одновременно дополнительной (функциональной) подчиненности. Наиболее ярким примером может служить бухгалтерия крупной компании, имеющей несколько направлений деятельности. Бухгалтеры, обслуживающие некоторое направление деятельности такой компании, входят в состав единой бухгалтерии и подчиняются (дисциплинарно) главному бухгалтеру (иногда финансовому директору). Однако функциональная подчиненность (в рамках основных функциональных обязанностей бухгалтеров, обслуживающих направление) подразумевает их подчинение руководителю функционального блока (направления).

### Структурный анализ организации

Характерной проблемой является наличие неофициальных отношений подчинения. Формально зафиксированное подчинение одних сотрудников другим на практике зачастую отсутствует. В результате появляется новая организационная структура, в целом соответствующая формальной, но в определенных частях отличающаяся от нее.

Третья серьезная проблема связана с отделением юридической структуры от управленческой. Эта особенность характерна в первую очередь для компаний-холдингов, имеющих в своем составе несколько юридических лиц. Управленческая структура (структура подчинения с точки зрения оперативного управления) почти всегда значительно отличается от юридической. Это объясняется тем, что существуют разные принципы и критерии формирования управленческой и юридической структур. Юридическая структура формируется с точки зрения интересов стратегического управления, а также с точки зрения требований бизнеса, которым занимается организация. Управленческая же структура выстраивается и оптимизируется с точки зрения более эффективного оперативного управления. В результате в одном подразделении (в рамках управленческой структуры) могут работать специалисты, состоящие в штате нескольких юридических лиц.

Структура территории может оказаться важной для распределенных организаций, где территориальное расположение отдельных подразделений (филиалов) в значительной мере влияет на особенности устройства системы управления, в частности, бизнес-процессами.

Несмотря на то, что во многих организациях нет четко сформулированных регламентных документов, описывающих правила ведения бизнеса и выполнения связанных с этим процедур, структуру основных и вспомогательных процессов верхнего уровня можно определить, и это должно быть сделано в самом начале работ по моделированию. Данная структура в той или иной степени идентична для всех компаний, занятых аналогичной деятельностью. В связи с этим можно использовать существующие обобщенные (референтные) модели процессов, создаваемые

для различных отраслевых областей. Выделение структур процессов обеспечит в дальнейшем более эффективное планирование и управление в ходе моделирования, а также облегчит получение структурированной информации о деятельности моделируемой организации.

Одной из важных задач повышения эффективности деятельности организации является оптимизация документооборота и создание системы управленческого учета. Для решения этой задачи необходимо иметь структурированную систему классификации всего информационного пространства организации, включающего как документы, так и отдельные экономические, финансовые, производственные и другие показатели. Формирование данной структуры — один из наиболее приоритетных этапов моделирования.

Задачи, связанные с созданием и внедрением информационных технологий, требуют детального анализа существующих информационных систем — их структуры и участия в бизнес-процессах организации. В связи с этим, необходимо заранее, до детального моделирования процессов, сформировать структурированный перечень всех интересующих информационных систем, а также оценить их внутреннюю структуру (прежде всего — набор основных модулей и экранных форм).

Таким образом, для того, чтобы построить адекватную и востребованную модель организации необходимо уже на первоначальных этапах моделирования задуматься о выделении и фиксации всех основополагающих структур. Грамотное их формирование обеспечивает качественный «задел» на будущее. Это позволит продуманно и прогнозируемо разработать все новые детальные модели, имеющие определенное место в общей модели структуры организации и соответствующие целям анализа отдельных элементов и организации в целом.

От качества и количества информации, полученной при изучении организации, зависит, насколько адекватной будет построенная модель.

Разработка моделей деятельности организации включает несколько этапов:

- выделение множества объектов, оказывающих существенное влияние на деятельность структурного элемента;
- спецификацию входных и выходных потоков (информации, материалов, продуктов, услуг, финансов и т.д.);
- выявление основных процессов, определяющих деятельность структурного элемента и обеспечивающих реализацию его целевых функций;
- спецификацию потоков между основными процессами деятельности, уточнение связей между процессами и внешними объектами;

#### **Моделирование бизнеса. Методология ARIS**

- оценку объемов, интенсивности и других необходимых характеристик потоков;
- разработку функциональной модели деятельности структурного элемента;
- объединение моделей структурных элементов в единую модель деятельности организации.

Построенная модель должна быть оптимизирована по критериям, представляющим интерес для пользователя. После этого проводится анализ моделей, результаты которого используются для реорганизации деятельности

## 3. Методология ARIS

### 3.1. Общие сведения.

#### Постановка задачи

Методология ARIS предполагает определенный подход к формализации информации о деятельности организации и представление ее в виде графических моделей, удобном для понимания и анализа. Модели, создаваемые по методологии ARIS, отражают существующую ситуацию с той или иной степенью приближенности. Степень детализации описания зависит от целей проекта, в рамках которого проводится моделирование. Модели ARIS могут быть использованы для анализа и выработки различного рода решений по реорганизации деятельности предприятия, в том числе по внедрению информационной системы управления, разработке систем менеджмента качества.

Методология ARIS реализует принципы структурного анализа и позволяет определить и отразить в моделях основные компоненты организации, протекающие процессы, производимую и потребляемую продукцию, используемую информацию, а так же выявить взаимосвязи между ними.

Создаваемые модели представляют собой документированную совокупность знаний о системе управления, включая организационную структуру, протекающие процессы, взаимодействия между организацией и субъектами рынка, состав и структуру документов, последовательность шагов процессов, должностные инструкции отделов и их сотрудников. В отличие от других подходов, методология ARIS предполагает хранение всей информации в едином репозитории, что обеспечивает целостность и непротиворечивость процесса моделирования и анализа, а также позволяет проводить верификацию моделей.

Преимущества методологии ARIS:

- возможность рассматривать объект с разных точек зрения; разные уровни описания, обеспечивающие поддержку

концепции жизненного цикла систем; дифференцированный взгляд на анализируемый объект (организацию, систему управления и т.д.);

- богатство методов моделирования, отражающих различные аспекты исследуемой предметной области, позволяет моделировать широкий спектр систем (организационно-хозяйственных, технологических и прочих);

- единый репозитории; все модели и объекты создаются и хранятся в единой базе проекта, что обеспечивает построение интегрированной и целостной модели предметной области;

- возможность многократного применения результатов моделирования; накопленное корпоративное знание о всех аспектах деятельности организации может в дальнейшем служить основой при разработке различных проектов непосредственно в среде ARIS и с использованием интерфейсов и других средств.

### 3.2. Предназначение системы ARIS. Примеры успешного использования

Инструментальная система ARIS 5.0, реализующая методологию ARIS, предназначена для визуального представления принципов и условий функционирования различного рода организаций, а также для анализа их деятельности по различным показателям. Целью такого анализа является определение идеальных характеристик, реформирование организационной структуры, функций, бизнес-процессов, используемых данных.

### Моделирование бизнеса. Методология ARIS

В рамках методологии ARIS имеется также возможность определить требования к автоматизированной системе управления и провести ее проектирование.

Круг компаний, использующих интегрированную инструментальную среду ARIS, достаточно широк. Организации, занимающиеся консалтинговой деятельностью, применяют ARIS как средство для анализа деятельности фирм-клиентов. Четыре из пяти ведущих мировых консалтинговых фирм используют ARIS в качестве инструментария для оптимизации своей деятельности. Различного рода компании, занимающиеся как производством продукции, так и предоставлением услуг, успешно применяют систему ARIS для анализа и реорганизации собственной деятельности и поддержания внутренних процессов при изменениях ситуации на рынке. Компания всегда имеет в своем распоряжении визуальное отображение всех аспектов своей деятельности, а также средства для их анализа (стоимостного, динамического, анализа соответствия организационной структуры выполняемым функциям и т.д.). Продукт ARIS может быть использован фирмами-разработчиками программного обеспечения для поддержки проектирования информационных систем.

Если деятельность предприятия поддерживается системой управления класса ERP, например, системой R/3 фирмы SAP, то комплекс ARIS будет постоянно обеспечивать такой системе актуальное состояние, соответствующее существующим на предприятии бизнес-процессам.

Направленность данного продукта на управление качеством по международному стандарту ИСО 9001 позволит компаниям вывести их деятельность на уровень, соответствующий общепринятым требованиям к организации процессов. Кроме того, данная система обеспечит поддержку процесса подготовки сертификации на соответствие данному стандарту. Для этого в системе предусмотрена возможность для создания специальных отчетов, отвечающих требованиям ИСО 9001, а также модели, отражающие полную методологию процесса сертификации. Правильное проведение всех необходимых мероприятий позволит

предприятию успешно выйти на европейский рынок и быть признанным большинством компаний Европы.

Система ARIS в настоящее время уже успешно используется множеством известных компаний различного профиля как в Европе, в том числе в России, так и по всему миру.

Фирма Mercedes-Benz применяет ARIS с 1995 г. для анализа и совершенствования деятельности в области производства легковых автомобилей. Поводом для ее приобретения послужила возрастающая конкуренция на рынке легковых автомобилей и острая необходимость в повышении конкурентоспособности предприятия. Система ARIS позволила фирме перейти на качественно новый уровень в организации производства.

Автомобилестроительная компания Volkswagen, успешно использовала систему ARIS для реализации эффективной программы лизинга в двух дочерних подразделениях — Volkswagen Leasing и Volkswagen Bank. Разработанная программа лизинга «LEASIS» в настоящий момент контролирует около 99 % всего лизингового рынка Германии. Весь процесс разработки данной программы был полностью осуществлен в системе ARIS и продолжался почти 5 месяцев при участии от 6 до 20 специалистов на разных этапах составления проекта. Окончательным этапом разработки стали расчет экономических показателей реализации программы, презентация проекта и обучение будущих участников данной программы, что также было организовано с применением инструментария ARIS.

В качестве подобных примеров можно привести и другие крупные и средние компании, занимающиеся производством товаров и предоставлением услуг.

### 3.3. Архитектура ARIS

Рассматриваемая методология ARIS [1,2] основана на разработанной профессором А.-В. Шеером теории «Архитектура интегрированных информационных систем» (Architecture of Integrated Information System — ARIS). Она определяет принципы моделирования практически всех аспектов деятельности организаций, что является ее коренным отличием от других методологий. Согласно терминологии, принятой в области структурного анализа, термин «архитектура» описывает типы используемых методов, их функциональные свойства и взаимоотношения между составными частями моделируемой системы [1].

Методология ARIS основывается на концепции интеграции, предлагающей целостный взгляд на бизнес-процессы, и представляет собой множество различных методологий, интегрированных в рамках единого системного подхода. Это позволяет говорить об общей архитектуре ARIS. К наиболее важным компонентам архитектуры ARIS относятся типы представления и уровни описания моделируемого объекта. Они рассмотрены далее.

Что обусловило выбор методологий и используемых в них моделей? Это прежде всего необходимость всестороннего описания сложной социально-технической системы, какой является практически каждая современная организация. В общем случае архитектура ARIS выделяет в организации такие подсистемы как:

- *Организационная.* Определяет структуру организации — иерархию подразделений, должностей и конкретных лиц, многообразие связей между ними, а также территориальную привязку структурных подразделений;

- *Функциональная.* Определяет функции, выполняемые в организации;

- *Подсистемы входов/выходов.*

Определяют потоки используемых и производимых продуктов и услуг;

- *Информационная (подсистема данных).*

Описывает получение, распространение и доступ к информации (данным);

- *Подсистема процессов управления.*

Определяет логическую последовательность выполнения функций посредством событий и сообщений. Можно сказать, что подсистема управления — это совокупность разнесенных во времени сообщений разного рода;

- *Подсистема целей организации.*

Описывает иерархию целей, достигаемых в ходе выполнения того или иного процесса;

- *Подсистема средств производства.*

Описывает жизненный цикл основных и вспомогательных средств производства;

- *Подсистема человеческих ресурсов.*

Описывает прием на работу, обучение и продвижение по службе персонала организации;

- *Подсистема расположения*

*организационных структур.* Описывает территориальное расположение организационных единиц.

Все эти подсистемы организации в реальности и в моделях должны быть связаны между собой (рис. 13). Методология ARIS дает возможность описывать достаточно разнородные подсистемы в виде взаимоувязанной и взаимосогласованной совокупности различных моделей, которые хранятся в едином репозитории. Именно взаимосвязанность и взаимосогласованность моделей являются отличительными особенностями методологии ARIS.

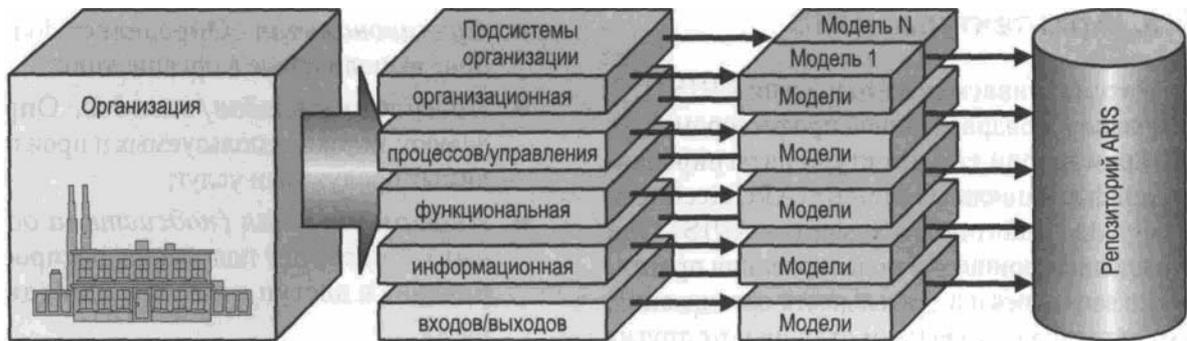


Рис. 13. Взаимодействие моделей в ARIS

В соответствии с правилами структурного анализа каждая из этих подсистем разбивается на элементарные блоки (модули), совокупность которых и составляет нотацию структурной модели той или иной подсистемы организации.

Естественно, что эти подсистемы не являются обособленными. Они взаимно проникают друг в друга, и поэтому одни и те же элементарные модули могут использоваться для описания различных структурных моделей. Для устранения избыточности методология ARIS ограничивает число типов моделей.

В связи с этим в методологии ARIS выделено четыре основных вида моделей, отражающих основные аспекты организации — пять типов представлений:

- *организационные модели*, описывающие иерархическую структуру системы — иерархию организационных подразделений, должностей, полномочий конкретных лиц, многообразие связей между ними, а также территориальную привязку структурных подразделений;

- *функциональные модели*, описывающие функции (процессы, операции), выполняемые в организации;

- *информационные модели (модели данных)*, отражающие структуру информации, необходимой для реализации всей совокупности функций системы;

- *модели процессов/управления*, представляющие комплексный взгляд на реализацию деловых процессов в рамках системы и объединяющие вместе другие модели;

- *модели входов/выходов*, описывающие потоки материальных и нематериальных входов и выходов, включая потоки денежных средств.

Остальные подсистемы могут моделироваться с использованием объектов, входящих в перечисленные выше типы представления. Графически такой подход представлен на рис. 14.

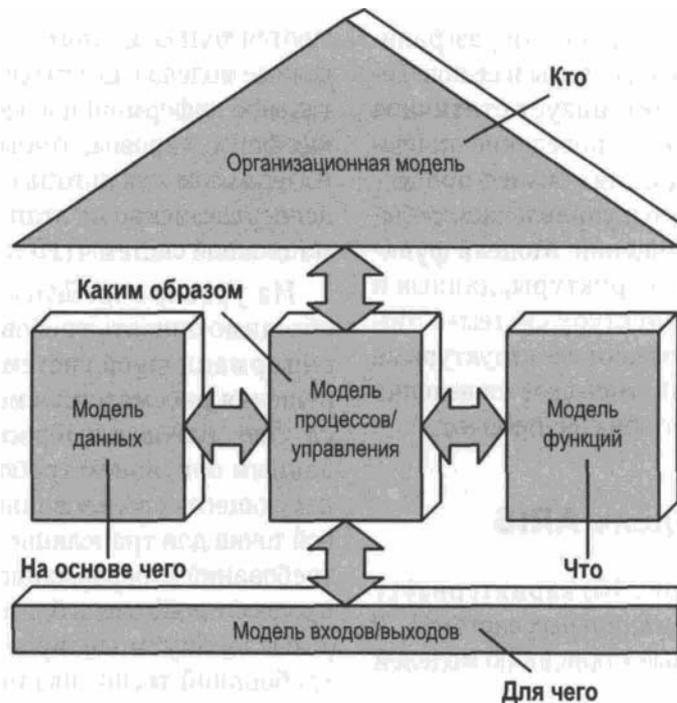


Рис. 14. Взаимосвязь видов моделей ARIS (здание ARIS)

Типы представления являются первой компонентой архитектуры. Они позволяют структурировать бизнес-процессы и выделять их составные части, что делает рассмотрение более простым. Применение этого принципа позволяет с различных точек зрения описывать содержание отдельных частей бизнес-процесса, используя специальные методы, наиболее полно соответствующие каждой точке зрения. Это избавляет пользователя от необходимости учитывать множество связей и соединений.

В рамках каждого типа представления создаются модели, отражающие ту или иную сторону исследуемой системы. Методология ARIS включает большое количество методов моделирования, в том числе известных как диаграммы Чена ERM, язык UML (Unified Modeling Language), методики OMT (Object Modeling Technique), BSC (Balanced Scorecard) и т.п.

Достоинство такого подхода заключается в том, что при анализе деятельности организации каждому аспекту можно уделять достаточное внимание, не отвлекаясь на его связь с другими аспектами. И только после детального изучения всех аспектов можно перейти к построению интегрированной

модели, отражающей все существующие связи между подсистемами организации.

Организационные и функциональные модели, а также модели данных, входов/выходов и процессов/управления рассматриваются как поля в специальной базе данных, называемой *репозиторием*. Репозиторий является ядром информационной системы, реализующей методологию ARIS. Он оказывает решающее воздействие на эффективность применения моделей. Интеграция различных видов моделей становится возможной благодаря хранению их в едином репозитории.

Методология ARIS не накладывает ограничений на последовательность подготовки пяти типов представления. Процесс анализа и проектирования можно начинать с любого из них, в зависимости от конкретных условий и целей, стоящих перед исполнителями.

В теории систем можно провести разграничение между структурой системы и ее поведением. Структура характеризует статичное представление системы, а поведение описывает ее динамику. В моделях бизнес-процессов динамика выражается управлением событиями и потоками сообщений. Модели функций, организационной структуры, данных и выходов описывают структуру системы. Модели управления показывают все структурные связи и описывают динамическое поведение потока, отображающего бизнес-процесс.

### 3.4. Фазовая модель ARIS

Фазовая модель (рис.15) характеризует этапы создания информационных систем [1,2] и подходы, применяемые к описанию моделей бизнеса. Согласно методологии ARIS модель организации структурируется в соответствии с концепцией жизненного цикла информационных систем, который представляется в виде последовательности уровней или этапов. Однако модель жизненного цикла ARIS не может рассматриваться как совокупность процедурных моделей для разработки некоторых независимых объектов на каждом уровне описания. Различные типы представления «выстроены» в модели в зависимости от степени их близости к информационным технологиям (ИТ).

Этот аспект реализуется посредством второй компоненты архитектуры ARIS, состоящей из различных *уровней описания*. Они дифференцируются по отношению к информационным технологиям. Такая концепция обеспечивает целостное описание управления бизнесом, вплоть до его технической реализации.

*Анализ проблем бизнеса* является начальной точкой при моделировании. Модели на этом уровне — не очень детальные семантические описания бизнес-процессов, однако они достаточно точно отражают цели, которые стоят перед разработчиками. На этом этапе в описание включаются характеристики будущей модели организации, связанные с бизнес-процессами. Для описания проблем

бизнеса используются только семантические модели. Полученные модели еще не содержат достаточно детальной информации и однозначных технических формулировок, чтобы служить исходным материалом для автоматической передачи их непосредственно на этап реализации информационной системы (ИС). На уровне *определения требований* необходимо описать требования к прикладной информационной системе, создаваемой для решения рассматриваемой проблемы бизнеса. Оно должно поддерживаться формализованным описанием требований с целью последующего использования в качестве стартовой точки для трансляции сформулированных требований в информационную систему. Этот процесс также очень близок к семантическому (смысловому) моделированию. Определение требований тесно связано с описанием проблем бизнеса. Эта связь показана широкой стрелкой на рис. 15.

Уровень *спецификации проекта* достигается, как только концептуальные понятия проблем бизнеса, сформулированные на уровне определения требований, трансформируются в категории, связанные с информационными технологиями. На данном уровне описываются уже не функции, а пользовательские или модульные транзакции, которые выполняют функции. Это может рассматриваться как отображение сформулированных требований в категории и методы описания, связанные непосредственно с информационными системами и выраженные в терминах соответствующих технологий. Таким образом, уровни определения требований и спецификации проекта связаны достаточно тесно.

Спецификация проекта может изменяться, не оказывая влияния на результаты предыдущего уровня определения требований. Однако это не означает, что определение требований и спецификация проекта могут прорабатываться независимо друг от друга. После того как завершён этап определения требований, его наиболее важная содержательная часть, отражающая категории управления бизнесом, должна быть определена таким образом, чтобы все, относящееся к области информационных технологий и программных решений

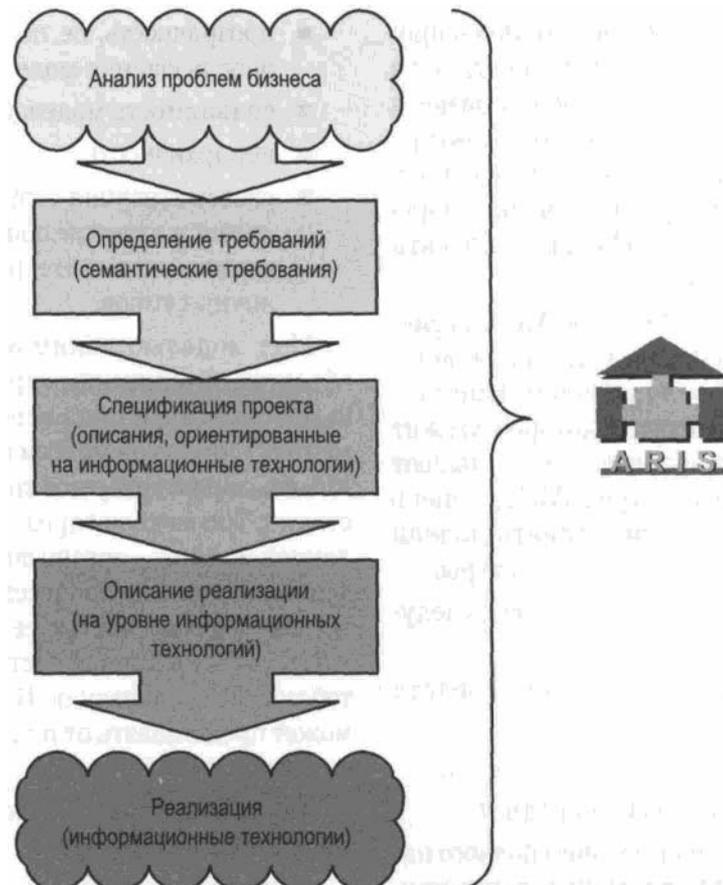


Рис. 15. Фазовая модель ARIS

(например, производительность информационной системы), не влияло на предметное содержание.

На уровне *описания реализации* спецификация проекта трансформируется в конкретные аппаратные и программные компоненты. Таким образом, осуществляется физическая связь с информационной системой.

Отдельные уровни описания имеют различные циклы корректировки. Частота корректировок выше всего на уровне описания реализации и ниже всего на уровне определения требований.

Уровень описания реализации очень тесно связан с разработкой информационной системы: на этом уровне производится многократная корректировка функционирования системы по результатам коротких циклов (тестов) ее работы.

Уровень определения требований особенно важен, поскольку его можно рассматривать как репозиторий для прикладных программных систем, используемых в течение длительного времени, и как стартовую точку при описании реализации. Документы, созданные на уровне определения требований, имеют наиболее продолжительный жизненный цикл, и благодаря их близости к описанию проблем бизнеса, которое также является документом, они чрезвычайно полезны для разработки информационных систем. По этой причине уровень определения требований, или семантическая модель, имеет наивысший приоритет. Семантические модели образуют связь между пользователями и первоначальным описанием их проблем на языке, ориентированном на категории информационных систем.

Создание различных видов моделей и проработка каждой из них по уровням описания в сочетании с формулировкой проблем бизнеса и составляет процесс работы в архитектуре ARIS. Каждый тип представления подвергается разложению на три уровня описания: определение требований, спецификацию проекта и описание реализации.

Таким образом, в архитектуре ARIS зафиксирован набор видов моделей, каждая из которых «расписывается» по уровням. Вместе с описанием проблем бизнеса, которое служит стартовой точкой для анализа, они составляют набор компонент архитектуры ARIS. Теперь необходимо выбрать и представить модели описания каждой компоненты архитектуры. Критерии для выбора этих моделей следующие:

- простота и выразительность средств изображения;
- поддержка смыслового содержания;
- отображение специфики предмета;
- возможность использования полного набора методов для различных типов приложений;
- степень знакомства с методами и наличие необходимой литературы;
- определенная степень независимости методов от технической реализации в информационных и коммуникационных системах.

### 3.5. Модели ARIS

При построении моделей методология ARIS требует соблюдения определенных принципов [1]. К ним относятся:

- корректность модели;
- релевантность (следует моделировать только те фрагменты реальной системы, которые соответствуют назначению системы, т.е. модель не должна содержать избыточной информации);
- соизмеримость затрат и выгод;

- прозрачность, т.е. понятность и удобство использования модели;
- сравнимость моделей;
- иерархичность;
- систематизация структуры, что предполагает в качестве обязательного условия возможность интеграции моделей различных типов.

Под моделью понимается совокупность объектов, объединенных друг с другом различными связями, и ряда вспомогательных элементов. Структура модели показана на рис. 16. Модель характеризуется типом, именем и свойствами. Тип модели определяет, что описывает данная модель — организационную структуру, функции, данные, процессы или выходы. Имя модели является частью ее атрибутов. Модель ARIS так же является частью более развернутой модели организации. В то же время модель может представлять отдельный объект, будучи его детальным описанием. Модели классифицируются с помощью методологических фильтров.

Каждая модель ARIS содержит:

- объекты — неделимые части модели, выделенные по какому-либо признаку, сформулированному в соответствии с методологией ARIS, и имеющие набор изменяемых характеристик — свойств, описывающих их поведение;
- связи между объектами — описанные взаимоотношения между объектами, имеющими свои свойства и характеристики. Так же как и объекты, связи характеризуются свойствами, выделены внешний вид и атрибуты.

Модель может включать:

- внешние встроенные объекты, например, рисунки, документы текстовых редакторов и т.п.;
- текст, размещенный в любом месте модели;
- геометрические фигуры.

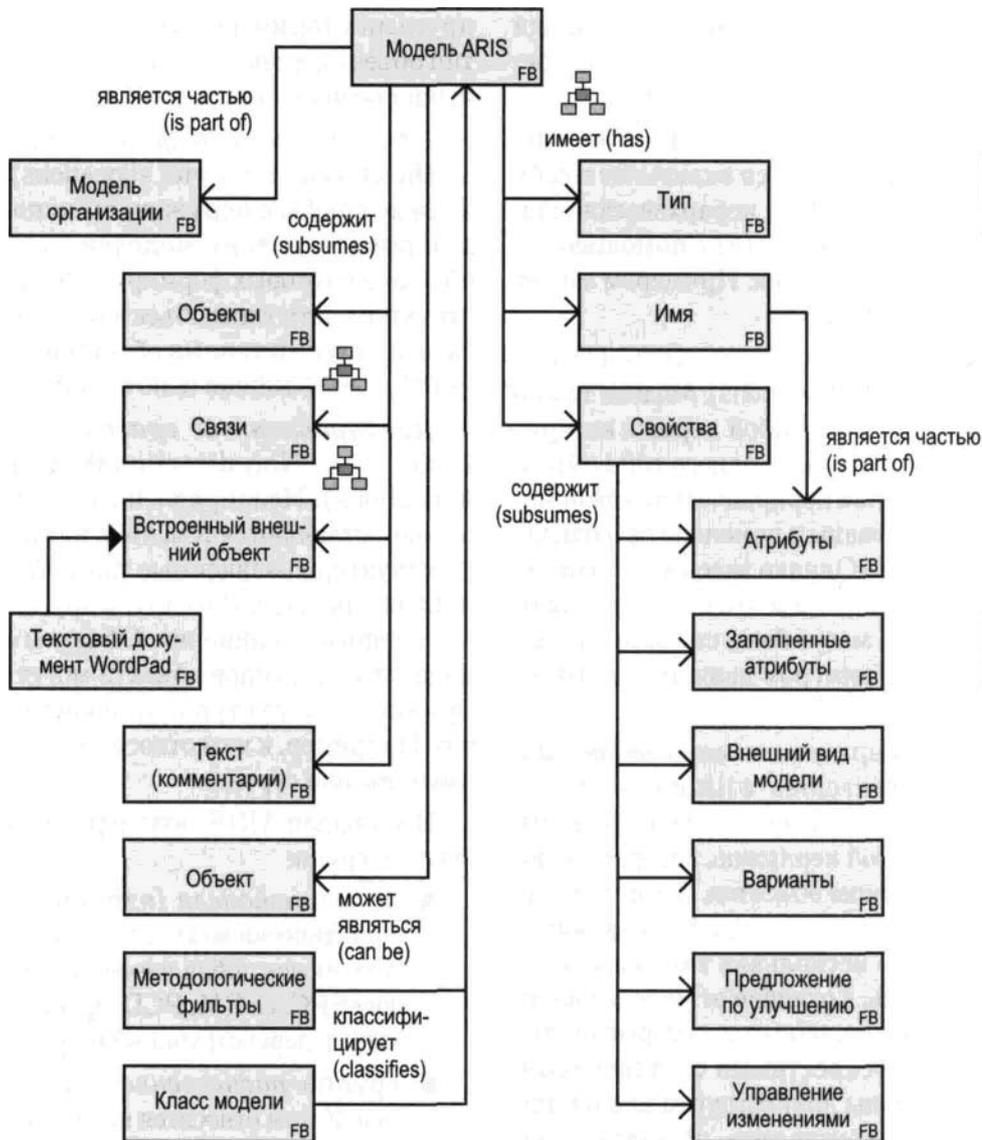


Рис. 16. Модель технических терминов, описывающая понятие «модель ARIS»

Каждая модель имеет ряд свойств:

- атрибуты (*Attributes*),
- заполненные атрибуты (*Maintained Attribute*),
- внешний вид модели (*Model Appearance*);
- варианты (*Variants*);

- предложения по улучшению (*Improvement proposals*),
- управление изменениями (*Change management*).

Каждая модель может содержать один или несколько десятков объектов. Число связей, возможных в данной модели, сильно варьирует.

### Моделирование бизнеса. Методология ARIS

Модели классифицируются при помощи методологических фильтров и классов.

Существуют следующие классы моделей:

*Строгая иерархия* (PH — Pure hierarchy).

Модели такого класса включают в себя только один тип объекта, а иерархические связи могут быть организованы с помощью нескольких типов соединения. Примером может служить дерево функций.

*Строгая иерархия с окружением* (HA — Pure hierarchy with allocations). Модели такого класса представляют собой дерево, которое строится на основе только одного типа объекта. В данном случае иерархические связи могут быть организованы с помощью нескольких типов соединения. Однако здесь, в отличие от строгой иерархии, с объектами, составляющими иерархию, могут быть связаны другие типы объектов. Примером является диаграмма целей.

*Иерархия с перекрестными соединениями* (HC — Hierarchies with cross links). Иерархия с перекрестными соединениями представляет собой иерархию, которая включает различные типы объектов. В данном случае иерархические связи могут быть организованы с помощью нескольких типов соединения. Однако здесь, в отличие от моделей класса строгой иерархии, объекты, которые входят в иерархию с перекрестными соединениями, могут быть связаны друг с другом как напрямую, так и через другие типы объектов. Примером является диаграмма структуры знаний.

*Тип основного объекта* (CO — Central object type). Модели данного типа используются для того, чтобы отобразить связи между этими типами объектов и другими типами объектов. Все типы объектов, прикрепленные к типу основного объекта, имеют одинаковые приоритеты. Примером служит диаграмма окружения функции, диаграммы ролей и прав.

*Приоритетный тип основного объекта* (CP — Central object type with priority).

Модели с приоритетным типом основного объекта используются для того, чтобы отобразить связи между типом основного объекта и

другими типами объектов. В данном случае тип объекта, имеющий наивысший приоритет, играет основную роль в иерархии.

*Направленный граф с окружением* (DQ — Directional graph with allocations).

Направленные графы с окружением используются для представления моделей, особые типы объектов в которых формируют направленную структуру. Эти типы объектов могут быть связаны с другими типами объектов. Например, eEPC — событийная цепочка процесса.

*Ненаправленный граф с окружением* (NGA — Non-directional graph with allocations).

Ненаправленные графы с окружением отличаются тем, что в них присутствуют структурно-зависимые типы объектов, однако, они не соединены друг с другом через направленные соединения. Объекты структурно-зависимых типов объекта могут иметь окружение из структурно-независимых объектов. Например, к ним относятся модели в виде столбцов или строк.

Все модели ARIS можно разделить на несколько групп:

- *основные модели* (ядро), используемые практически во всех проектах. Это модели цепочек добавленных значений, модели eEPC, eERM, PCD, организационная схема, дерево функций и другие;
- группа *расширение основных моделей*. К ним относятся карта знаний, диаграмма структуры знаний, диаграмма окружения функции и другие;
- группы, ориентированные на отдельные области моделирования'.
  - документация плана развития;
  - объектно-ориентированное моделирование;
  - анализ (создание индивидуальных референтных моделей и диаграмм топологии);
  - описание потока материалов;
  - внедрение системы SAP R/3 и использование ARIS Analyzer for R/3;

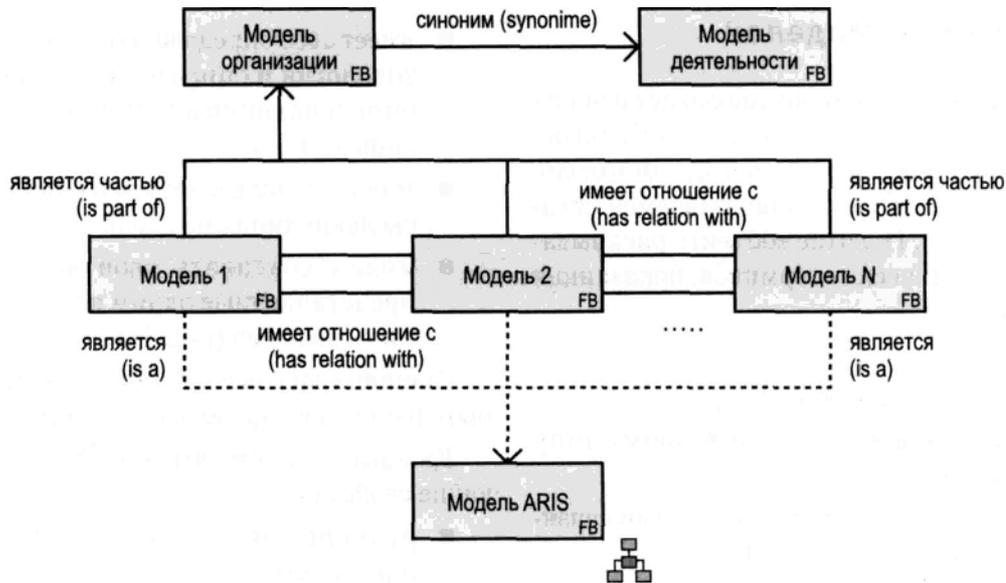


Рис. 17. Упрощенное представление модели организации

- моделирование workflow;
- варианты методов моделирования данных;
- модели для функционально-стоимостного анализа;
- модели для управления проектом.

Группа моделей ARIS, взаимосвязанных посредством использования общих объектов, называется *моделью организации* (моделью бизнеса) (рис. 17). Эта взаимосвязь осуществляется благодаря репозиторию, где храниться вся информация об объектах и их связях.

Преимущества ARIS как среды для моделирования бизнеса заключаются именно в существовании взаимосвязей между отдельными моделями, описывающими различные стороны такого многогранного явления, как деятельность организации.

Разработка моделей организации не является самоцелью. Они создаются для того, чтобы получить новую информацию о деятельности организации. Для ее получения методология ARIS предусматривает целый комплекс операций, которые могут быть проведены над моделями. Эти операции делятся на основные и вспомогательные (служебные).

К основным операциям относятся:

- проверки корректности моделей (семантические проверки);
- составление разнообразных отчетов по модели;
- оптимизация моделей по различным критериям;
- анализ моделей, проводимый по различным методикам, например, функционально-стоимостной анализ, стратегическое планирование;
- сравнение моделей;
- обмен информацией с другими программными системами;
- непрерывное улучшение модели.

Вспомогательные операции обеспечивают выполнение основных операций. К ним относятся:

- копирование моделей;
- создание вариантов;
- слияние моделей;
- экспорт/импорт моделей;
- хранение моделей, в том числе в виде резервных копий;
- очистка (консолидация) моделей

### 3.6. Объекты моделей

Каждый тип модели предполагает использование определенного набора объектов. *Объект* — самостоятельная часть методологии, отражающая элемент описываемой предметной области. Понятие «объект» раскрывает модель технических терминов, показанная на рис. 18.

Каждый объект:

- имеет уникальное имя;
- принадлежит к определенному типу объектов;
- соединен одной или несколькими связями с другими объектами;

- имеет свое определенное значение в методологии и описывается свойствами, определяющими конкретный объект данного типа;
- может использоваться в одной или нескольких типах моделей;
- может создавать свои экземпляры, представленные одним или несколькими символами (рис. 19).

Внешний вид экземпляра объекта может быть изменен в определенных пределах.

Каждый объект моделей ARIS имеет следующие свойства:

- размещение атрибутов (*Attribute placements*),

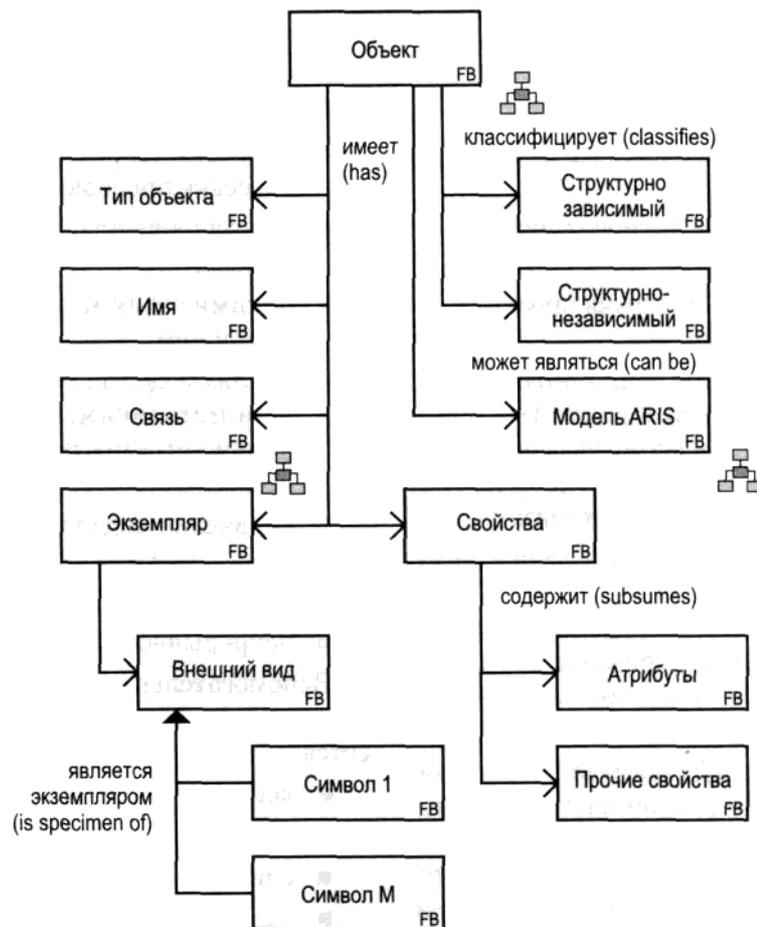


Рис. 18. Диаграмма технических терминов, описывающая понятие «объект»



Рис. 19. Различные изображения объекта «Организационная единица»

- взаимосвязи (*Relationships*);
- назначение (*Assignments*);
- местонахождение (*Occurrences*);
- внешний вид (*Object Appearance*);
- варианты (*Variants*);
- управление изменениями (*Change Management*);
- предложения по улучшениям (*Improvement Proposals*);
- заполненные атрибуты (*Maintained Attributes*);
- атрибуты объекта (*Attributes*).

Объекты классифицируются на:

- *структурно-зависимые*. Эти типы объектов определяют общую структуру модели. Конкретный тип модели подразумевает наличие конкретных структурно-зависимых типов объектов;
- *структурно-независимые*.

Методология ARIS предусматривает возможность описывать объекты более подробно с помощью детализирующих моделей. Такая связь между типом объекта (базовым объектом) и типом модели называется *типом детализации*. Отдельный экземпляр типа детализации, т.е. связь между определенным объектом и определенной моделью, называется *детализацией*. Типы детализации, разрешенные к использованию в конкретном случае, зависят от типа объекта, к которому принадлежит базовый объект. При создании детализации пользователь может создавать связи между базовым объектом и одним или большим количеством объектов из детализированной модели. Такая связь

между типами объектов, пересекающая границу между моделями, называется *типом детализирующей связи*. Отдельная связь между двумя объектами называется *детализирующей связью*.

Выбор типов связей и детализации позволяет управлять принципами и глубиной декомпозиции, а также «загруженностью» моделей.

При создании нового объекта в модели в базе данных ARIS появляется отдельная запись, задающая описание объекта. Типичные описания объекта включают атрибуты, которые характеризуют объект. Описания объектов должны быть уникальными по отношению к их именам. Таким образом, когда описания заданы, всегда производится проверка, не существует ли уже тот или иной объект в базе.

Экземпляр объекта всегда связан с отдельной моделью. Описание объекта всегда связано с отдельной группой. На основании существующего описания экземпляры данного объекта могут использоваться в других моделях. При изменении характеристик одного из экземпляров объекта меняются свойства всех остальных экземпляров и соответственно его описание в хранилище.

При удалении одного из экземпляров объекта все остальные остаются неизменными. В то же время при удалении описания объекта из хранилища уничтожаются все существующие его экземпляры.

В моделях ARIS используется большое число различных типов объектов (см. в разделе 12.2), и, соответственно, самих объектов значительно больше, поскольку один тип содержит, как правило, несколько объектов.

### 3.7. Связи объектов

Важную роль при моделировании играет правильный выбор связей между объектами модели. *Связь* — описание взаимоотношений между двумя объектами, имеющая определенные свойства (рис. 20). Каждая связь:

- имеет уникальное имя;
- принадлежит к определенному типу. *Тип связи* зависит как от типов соединяемых объектов, так и от направления соединения. Между некоторыми типами объектов связи не могут существовать;
- соединяет исходный и конечный объекты;
- характеризуется своими свойствами;
- может создавать свои экземпляры, внешний вид которых может быть изменен в определенных пределах.

Среди связей различают *соединения* и *детализирующие связи*.

*Соединения* — это такие связи, экземпляры которых представляют собой видимые соединения между двумя объектами внутри одной модели. Соединения классифицируются на структурно-зависимые и структурно-независимые. При детализации можно создавать связи между базовым объектом и одним или большим количеством объектов из детализирующей модели. Связь между типами объектов, пересекающая границу между моделями, называется *типом детализирующей связи*. Отдельная связь между двумя объектами называется *детализирующей связью*.

Каждая связь в рамках методологии ARIS имеет следующие свойства:

- размещение атрибутов (*Attribute placements*);
- внешний вид связи (*Connection Appearance*);
- местонахождение связи (*Connection Occurrences*);
- атрибуты связи (*Attributes*);
- заполненные атрибуты (*Maintained Attributes*);

Между двумя объектами может быть создана одна или несколько связей различных

типов. В рамках одной модели число типов связей может меняться от единиц (например, три — в дереве функций) до нескольких десятков (например, 24 — в организационной схеме). Один и тот же тип связи может присутствовать в нескольких моделях.

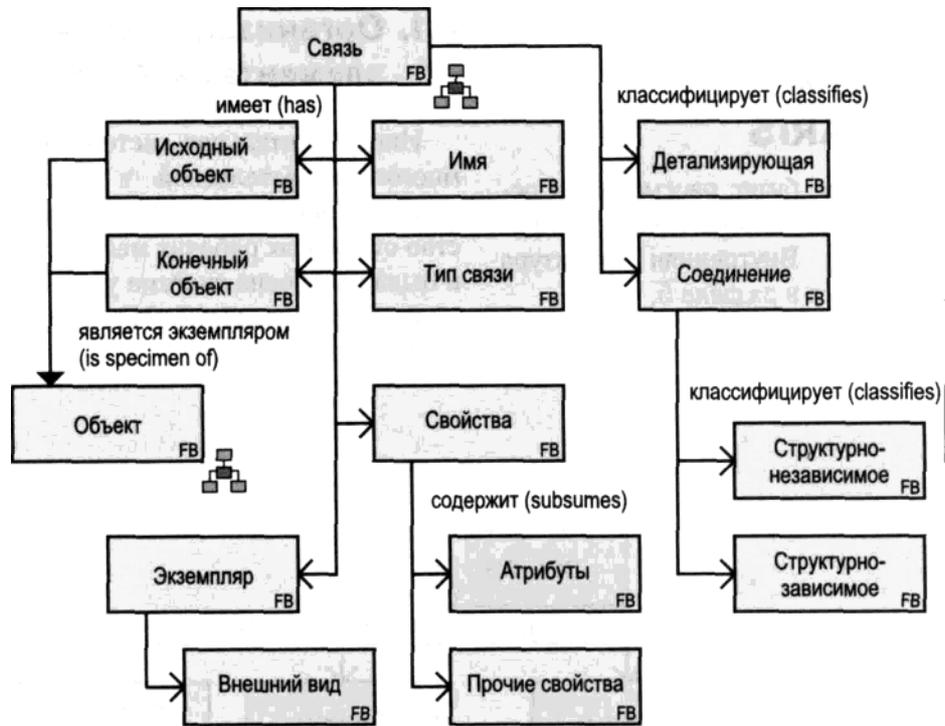


Рис. 20. Модель технических терминов, описывающая понятие «связь»

## 4. Основные элементы реализации системы ARIS

В данном разделе будут рассмотрены основные принципы реализации ARIS как программной системы. Внутренняя структура ARIS 5.0 описывается в разделе 5.

### 4.1. Организация управления элементами ARIS

Инструментальная система ARIS является многопользовательской и может работать в компьютерной сети, объединяющей множество отдельных рабочих мест. Исходя из этого и была построена система управления информацией о моделях (рис. 21) — сеть ARIS (*ARIS Network*).

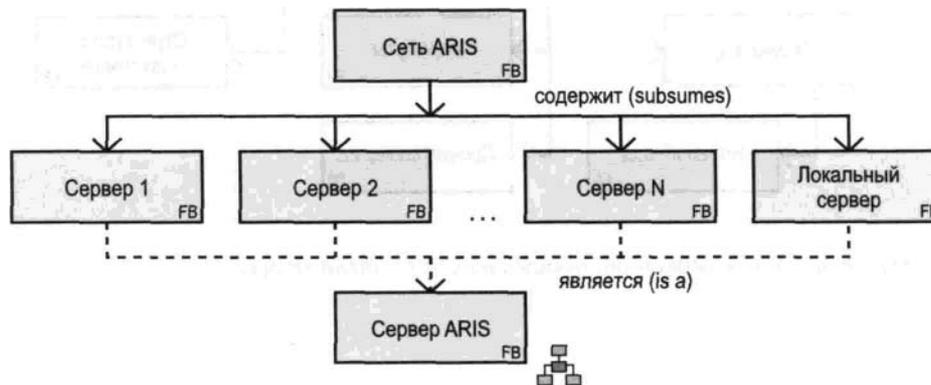


Рис. 21. Структура сети ARIS

Сеть ARIS является главной папкой (структурный элемент MS Windows) структуры, отображаемой в проводнике (*ARIS Explorer*). Сеть включает в себя все серверы баз ARIS, к которым имеет доступ клиент ARIS. Вместо имени *ARIS Network* главной папке можно дать любое другое имя. Через данный элемент устанавливается связь со всеми остальными серверами баз ARIS.

После установки других модулей ARIS, таких как *ARIS for mySAP.com*, сети, например, сеть SAP, также могут быть отображены в дереве каталогов проводника ARIS — в дополнение к сети ARIS. С технической точки зрения, термин *возможности многопользовательской работы* подразумевает, что локальные компьютеры (клиенты), на которых установлена система ARIS (ПО клиента),

соединены сетью с одним или несколькими компьютерами — серверами баз ARIS. Все данные пользователей обрабатываются системой, установленной на этих серверах. Каждый клиент ARIS имеет доступ к данным (т.е. моделям, именам моделей, именам папок, структуре папок, описаниям объектов, атрибутам и т.д.), которые хранятся на одном или нескольких серверах. Чтобы пользователь мог изменить или просмотреть эти данные, они должны быть считаны с сервера, переданы по сети, а затем обработаны и отображены ПО клиента. Точно так же измененные данные могут быть переданы от клиента к управляющей системе базы и сохранены в ней.

### Основные элементы реализации системы ARIS

Многопользовательский режим работы означает, что несколько пользователей могут работать с базой одновременно. Каким образом пользователь может получить доступ к определенным моделям, описаниям объектов и экземплярам объектов (и может ли он вообще получить к ним доступ), зависит от действий самого пользователя и взаимодействия отдельных компонентов ARIS.

В большинстве случаев при многопользовательском режиме работы ни один пользователь не может эксклюзивно войти в базу, так как это закрывает доступ к ней другим пользователям.

Временно закрытыми для записи могут оказаться лишь отдельные объекты базы — это происходит в тот момент, когда один из пользователей вносит или записывает в базу изменения, связанные с этими объектами. Данное решение значительно сокращает временные задержки, возникающие при конфликтах между системами различных пользователей.

Право вносить изменения в модель на уровне экземпляров дается только первому пользователю, открывшему эту модель. Так, второй пользователь, который открывает ту же модель, не может перемещать объекты внутри нее или вносить изменения свойства модели. Однако этот пользователь может переименовать описания объектов с помощью проводника и редактировать атрибуты соответствующих описаний объектов с помощью модуля *ARIS Attributes* (см. раздел 5.4).

Чтобы исключить параллельное редактирование, атрибут, изменяемый пользователем, становится защищенным от записи для других пользователей. Только первый пользователь, который открыл атрибут для редактирования, является единственным пользователем,

которому дается право вносить изменения в этот атрибут. Редактируемые атрибуты остаются защищенными от записи до тех пор, пока внесенные данным пользователем изменения не будут сохранены.

Основной структурной единицей сети ARIS является сервер. *Локальным* называется сервер, расположенный на конкретном рабочем месте. Он обладает всеми свойствами обычного сервера. ARIS позволяет иметь практически неограниченное число серверов. Структура сервера ARIS показана на рис. 22.

Сервер включает в себя:

- *метаданные* ARIS, состоящие из *метамodelей* ARIS — описаний всех моделей, доступных пользователю при моделировании, и конфигурационной базы, которая хранит информацию об используемых на данном сервере методологических фильтрах, шаблонах моделей, используемом формате шрифтов и описания диаграмм;
- неограниченное число баз данных, содержащих информацию о модели организации (или ее части).

За управление серверами отвечает модуль *ARIS Administration*, рассмотренный в разделе 5.11.

Приведенные выше понятия отображаются на левой панели одного из основных модулей ARIS — его проводника (рис. 23).

Любой из серверов содержит одну или несколько баз данных ARIS. Понятие «база данных» раскрывает рис. 24. Каждая база данных отображает модель деятельности организации (бизнеса). Она содержит информацию двух видов — вспомогательную (сервисную) и основную.

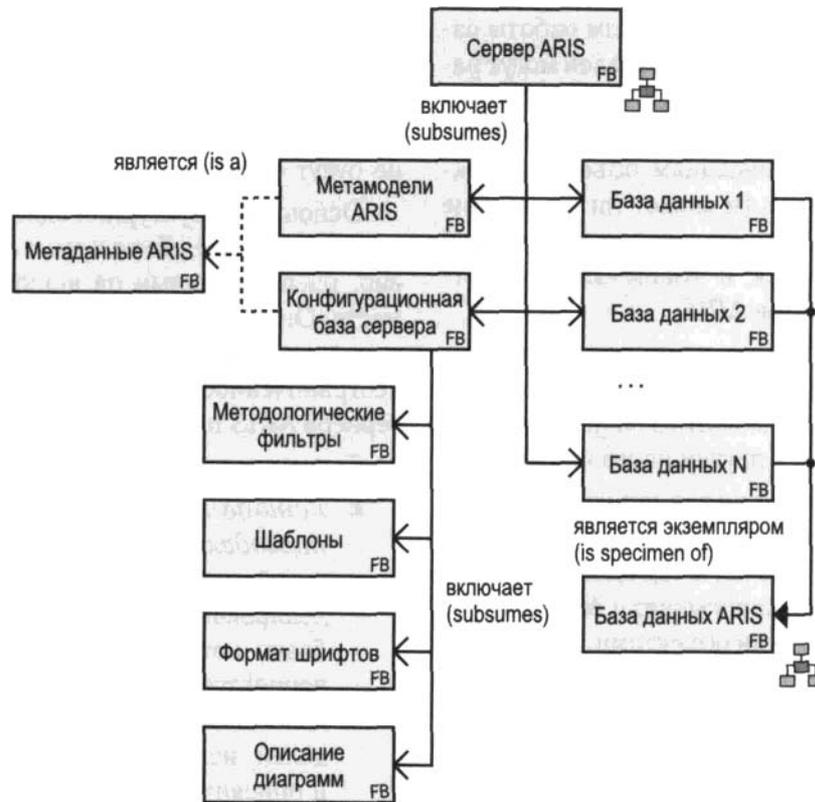


Рис. 22. Модель технических терминов, описывающая понятие «сервер ARIS»

## Основные элементы реализации системы ARIS

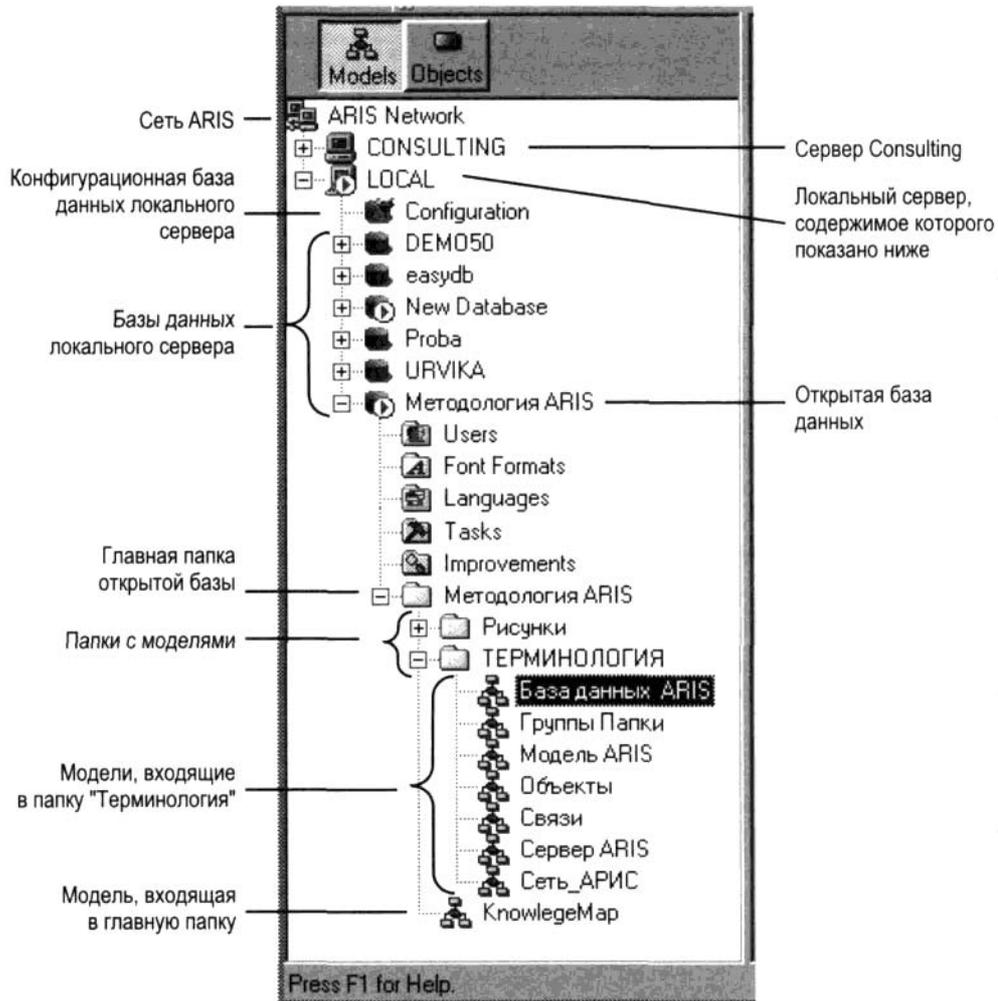


Рис. 23. Левая панель проводника ARIS Explorer

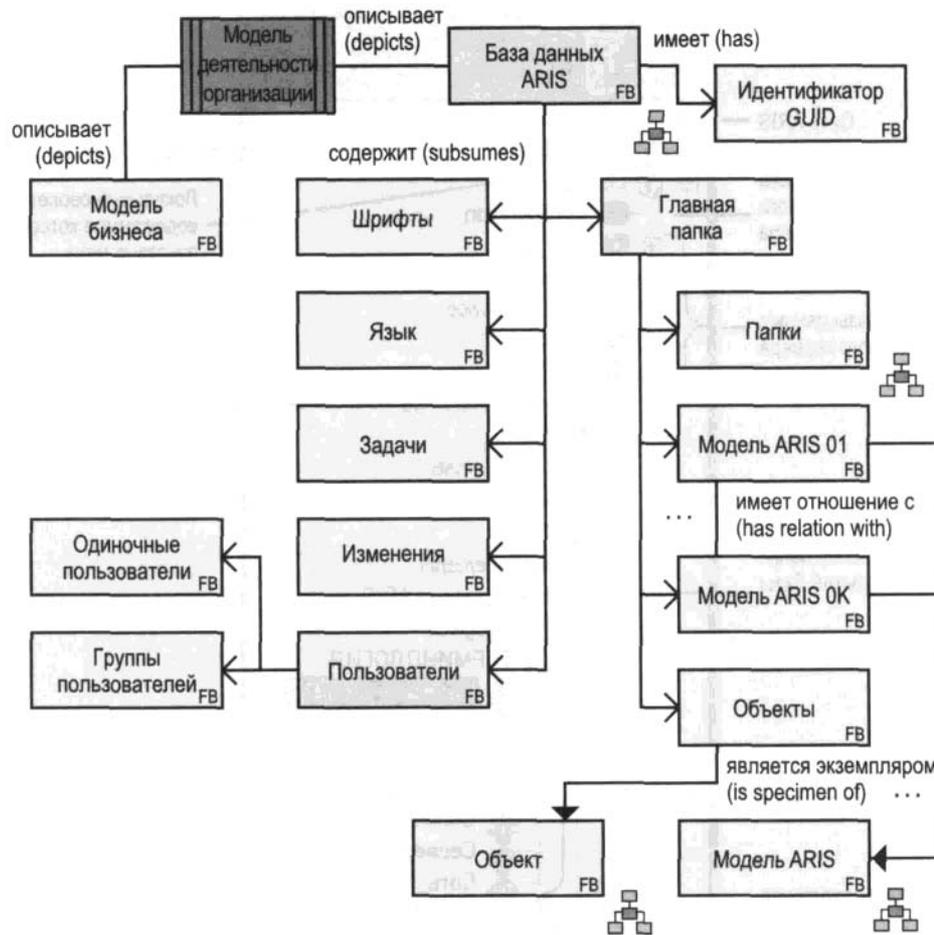


Рис. 24. Модель технических терминов, описывающая понятие «база данных» ARIS

К вспомогательной информации относятся папки, содержащие:

- сведения о пользователях (*Users*) и их группах, а также соответствующие средства управления. Пользователь или их группа характеризуются набором изменяемых свойств (см. раздел 9.8). Пользователи в базе ARIS могут быть созданы администраторами сервера конфигурации или базы, или пользователем, у которого есть права на управление пользователями. Каждому пользователю присваиваются собственные (уникальные) имя и пароль. Пользователи могут быть распределены по группам пользователей. В зависимости от принадлежности к той или иной группе пользователю могут быть доступны методологические фильтры и присвоены функциональные права и права доступа — в дополнение к тем, которые были присвоены ему изначально как отдельному пользователю

(индивидуально). Другими словами, пользователь получает в свое распоряжение методологические фильтры, функциональные права и права доступа, которыми обладает эта группа. Он лишается этих привилегий лишь в том случае, если их лишается вся группа. Подробно эти вопросы рассмотрены в разделе 9.8;

- сведения об установленных шрифтах (*Font Format*) и средства для их изменения и дополнения. Каждый установленный шрифт имеет корректируемые свойства:

### Основные элементы реализации системы ARIS

- информацию об используемых в моделях языках (*Languages*) и средства для их изменения и дополнения. Установленный язык описывается набором изменяемых свойств;
- задачи (*Tasks*), поставленные пользователями моделей в атрибутах *Change Managements* (*Управление изменениями*) (см. раздел 5.8);
- изменения, предложенные пользователями моделей в атрибутах *Change Managements*,

К вспомогательной информации относится и глобальный идентификатор GUID (*Global Unique Identifier*) базы данных, который существует также для таблиц, моделей, объектов и соединений и присваивается им при создании. Идентификатор GUID можно увидеть с помощью закладки *Maintained Attributes*

(*Заполненные атрибуты*) в тех диалоговых окнах, которые управляют свойствами баз, таблиц, моделей и т.п. GUID содержит строго 36 символов и имеет структуру 8-4-4-4-12, считая дефис за знак (см. рис. 25).

Основная информация базы данных содержится в главной папке (по аналогии с папками Windows), которая включает:

- произвольное число папок, создаваемых пользователем;
- произвольное число моделей и объектов ARIS, также создаваемых пользователем.

В этих структурах содержится то, что обычно называют моделью деятельности организации (рис. 17), хотя надо четко уяснить, что только одни модели без вспомогательной, сервисной, информации не могут полностью описать моделируемый объект.

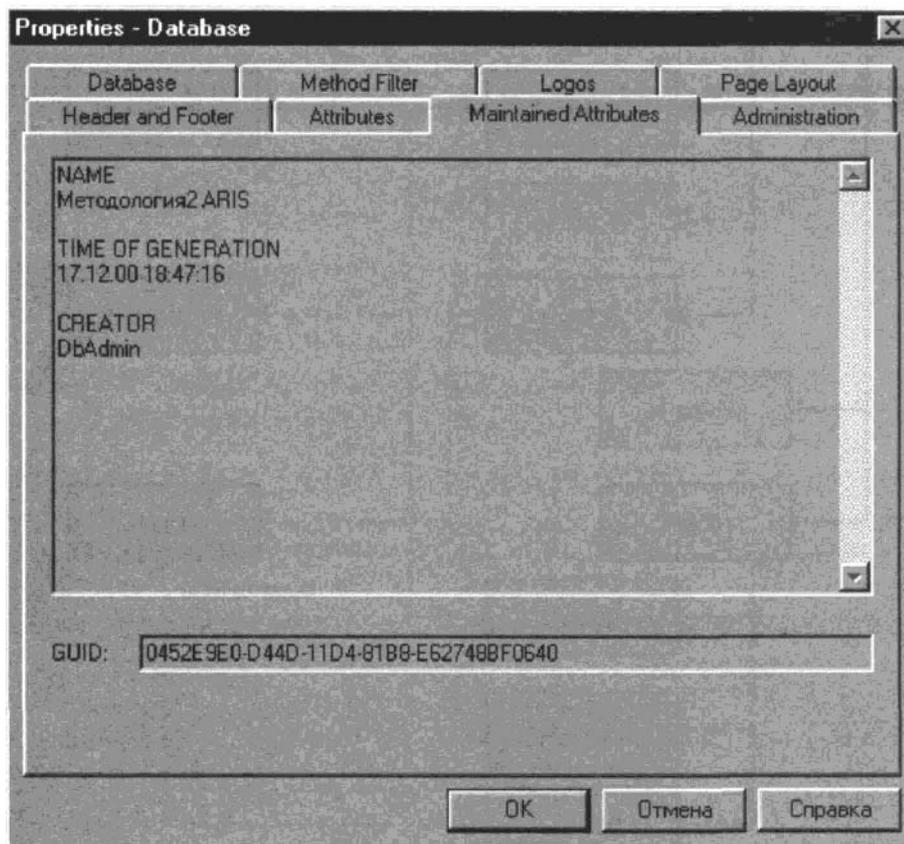


Рис. 25. Окно свойств базы данных

Понятие «папки» раскрывается на рис. 26. Папки используются в пользовательской базе ARIS для создания иерархической структуры для хранения моделей. Структура папки в базе сравнима со структурой папок в Проводниках Windows NT и Windows.

Папка верхнего уровня называется *главной папкой*. Она является корневой в структуре базы ARIS и не может быть удалена. Во всех других отношениях главная папка может быть отредактирована, как любая другая папка. Можно создать любое число папок, подчиненных главной папке. Они, в свою очередь, также могут иметь подчиненные папки.

Папки, как и другие элементы ARIS, обладают определенными свойствами, а именно привилегиями доступа (*Access Privileges*), атрибутами и заполненными атрибутами (рис. 27).

В папках размещаются модели и объекты ARIS. Эти понятия обсуждались в разделах 3.5 и 3.6. Модели разделяются в соответствии с пятью типами представления и тремя уровнями описания. Это принято представлять в виде здания ARIS (рис. 14). Однако при практической реализации методологии ARIS модели процессов/управления и модели входов/выходов были объединены, и здание ARIS приобрело вид, показанный на рис. 28.

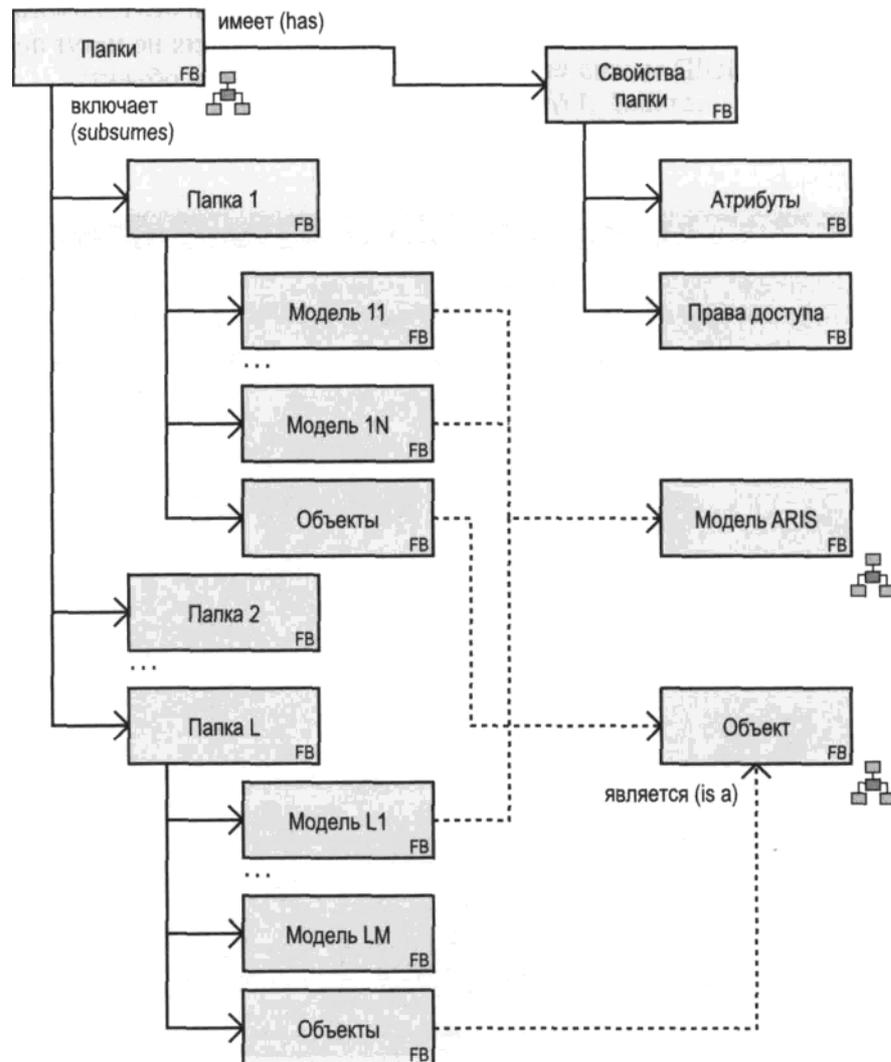


Рис. 26. Модель технических терминов, описывающая понятие «папка»

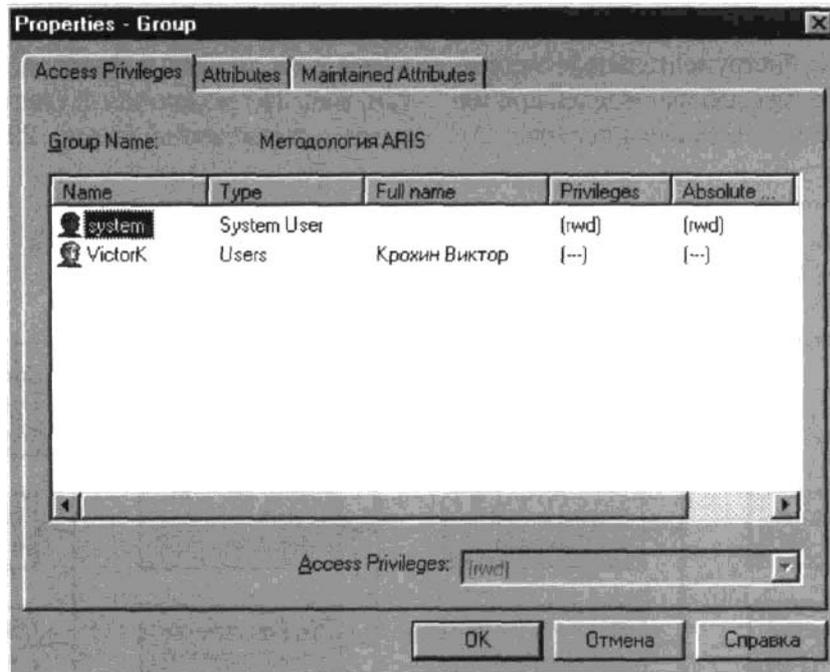


Рис. 27. Окно Properties Group (Свойства папки). Закладка Access Privileges (Привилегии доступа)



Рис. 28. Здание ARIS

Здание ARIS играет важную роль как в методологии, так и в инструментальной системе. Оно используется для выбора моделей при моделировании в модуле *ARIS Designer* (рис. 48)

Классификацию моделей ARIS в соответствии с их типами и фазовой моделью ARIS (см. рис. 15) можно представить в виде диаграммы, приведенной на рис. 29.

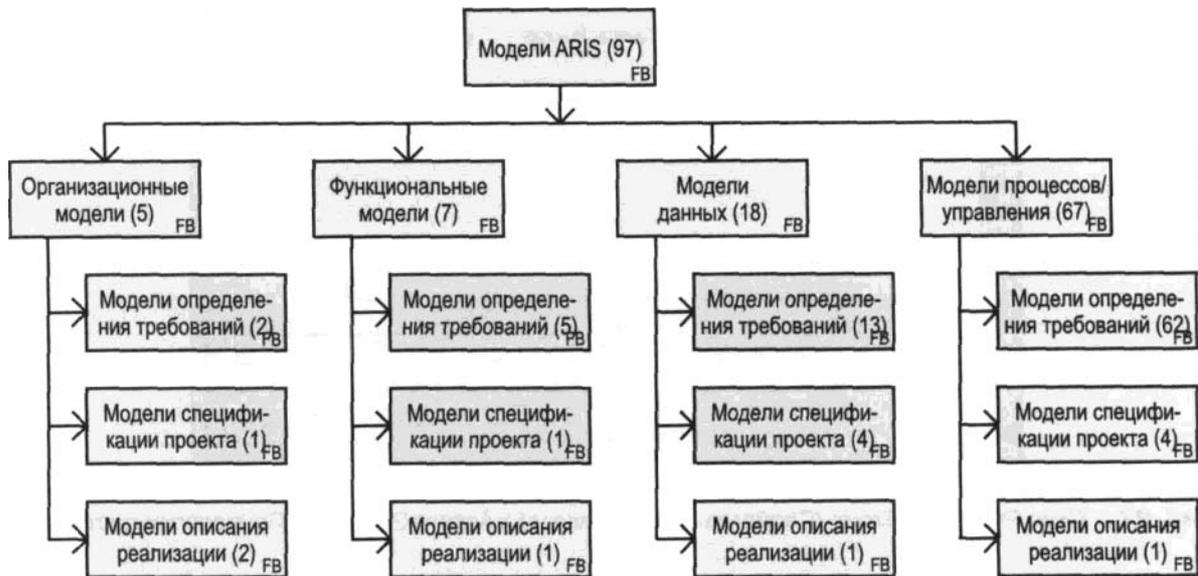


Рис. 29. Классификация моделей ARIS (в скобках показано число методов, реализованных в инструментальной системе ARIS 5.0 для каждого типа модели)

Как уже было сказано в главе 3, модели состоят из объектов и объединяющих их связей. Все эти элементы, также как и базы данных, имеют глобальный идентификатор GUD.

## 4.2. Атрибуты элементов ARIS

Каждый элемент ARIS (кроме серверов) обладает собственным набором атрибутов. Напомним, что атрибутом называется необходимое, существенное, неотъемлемое свойство объекта.

Атрибуты организованы в иерархическую древовидную структуру папок, аналогичную структуре «Сеть ARIS». Родственные по смыслу типы атрибутов собраны в папки. Состав папок зависит от типа модели, объекта или связи. Примеры атрибутов для модели, объекта и связи приведены на рис. 30—32.

Управление атрибутами элементов производится при помощи модуля *ARIS Attributes* (см. раздел 5.4).

Имеют атрибуты следующие элементы ARIS:

- пользователи (*User*);
- группы пользователей (*User Group*);
- папки (*Groups*);
- базы данных (*Databases*);
- модели (*Models*);
- объекты (*Objects*);
- связи (*Relationships*);
- языки (*Languages*);
- шрифты (*Font formats*);
- текст произвольной формы (*Free Form Text*);

### Основные элементы реализации системы ARIS

Число атрибутов, особенно для объектов, весьма велико и может достигать нескольких десятков. Выбор типов атрибутов зависит от решаемых задач. Например, при оптимизации продолжительности выполнения процессов необходимо задавать значения времени для выполнения отдельных функций. В других проектах этот атрибут использовать не имеет смысла.

Для того чтобы при моделировании получить понятные и модифицируемые результаты, необходимо решить, какие типы атрибутов будут заполнены в моделях, объектах и связях. На выбор тех или иных типов атрибутов в значительной степени влияют цели моделирования.

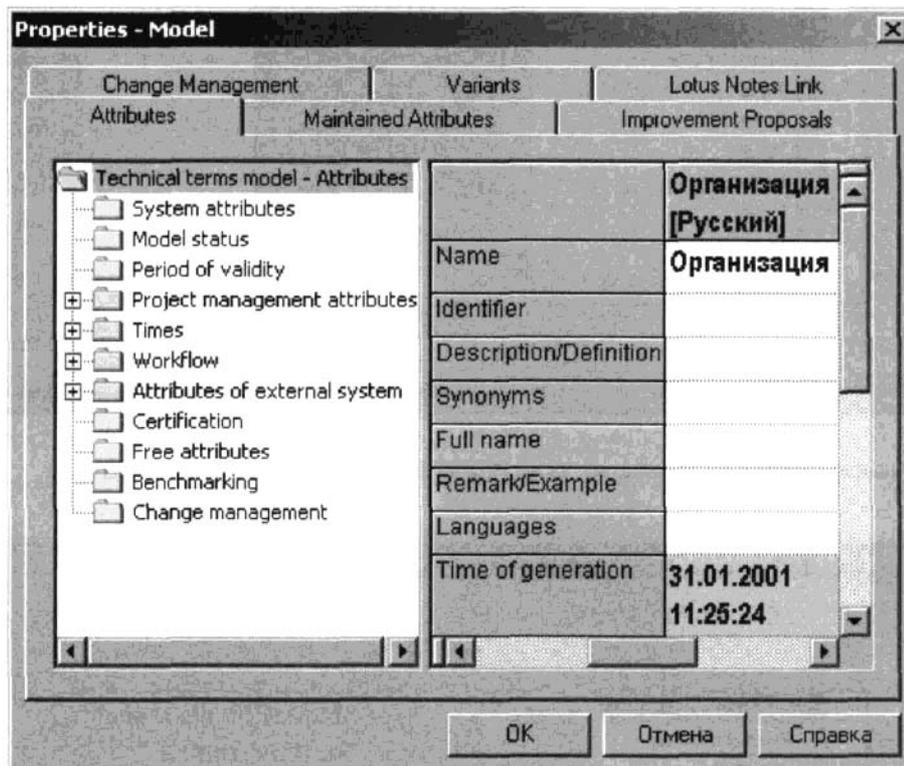


Рис. 30. Закладка Attributes (Атрибуты), окно Properties-Model (Свойства модели)

В ARIS информация о модели содержится в атрибутах этой модели (рис. 30). В зависимости от типа модели, можно заполнять различные атрибуты. Помимо имени модели, которое должно присутствовать обязательно, реко-

мендуется также для соответствующей модели заполнять атрибут «описание» {Description/Definition}, Кроме того, исходя из целей конкретного проекта, рекомендуется заполнять специальные атрибуты.

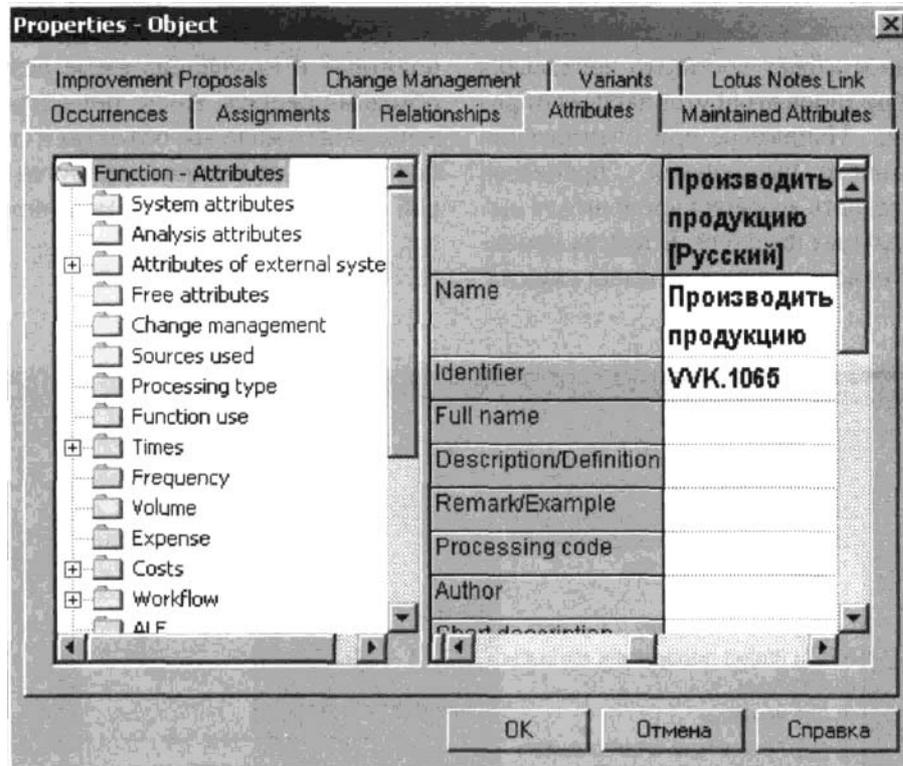


Рис. 31. Закладка Attributes (Атрибуты), окно Properties-Object (Свойства объекта)

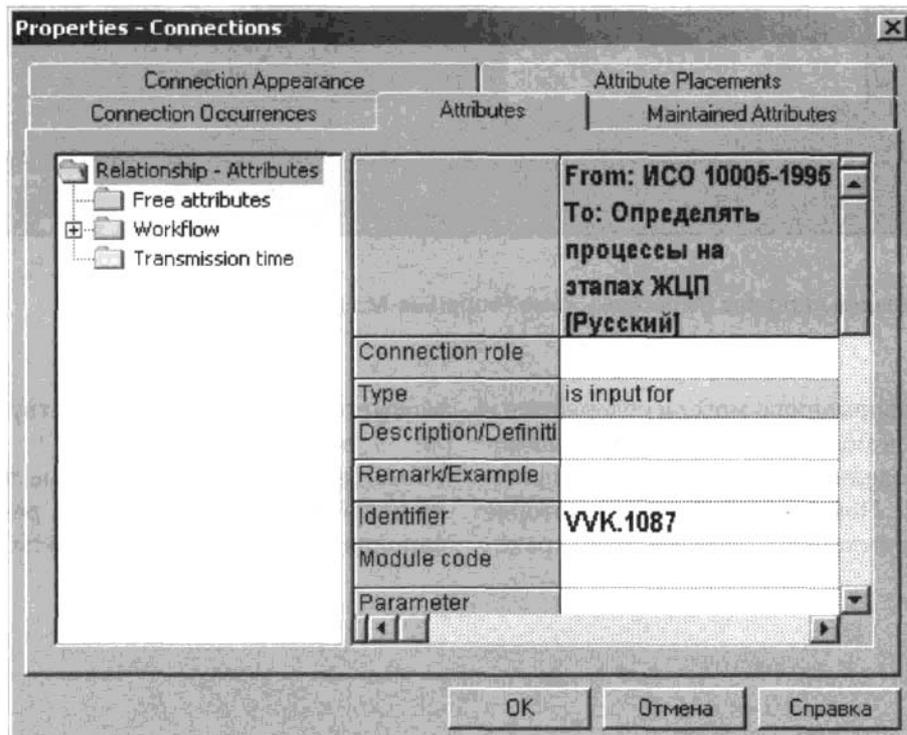


Рис. 32. Закладка Attributes (Атрибуты), окно Properties-Connections (Свойства связи)

Отметим особенности некоторых типов атрибутов, содержащихся в главной папке.

- *Name*. Имя рекомендуется указывать обязательно для всех элементов. Исключения составляют только «вспомогательные объекты», такие как *operators/rules* (*операторы/правила*), например, XOR, AND, и т.д., или *generalization/specialization* (*обобщение/специализация*) в моделях данных. Атрибут *Name* используется для выделения объекта (при создании копии, поиске, объединении объектов). Уникальность имени имеет очень важное значение. Имя и его уникальность очень важны при объединении баз данных (см. модуль *ARIS Merge* в разделе 5.15). В этом случае имя может использоваться как идентификатор объекта, т.е. как некоторая альтернатива атрибута *Identifier* (*идентификатор*).

Тем не менее система поддерживает возможность создания нескольких объектов одного типа с одинаковыми именами, что иногда является необходимым с точки зрения особенностей проводимого проекта.

- *Identifier*. Идентификатор используется в больших базах данных, чтобы избежать проблем с синонимами/омонимами в

#### Основные элементы реализации системы ARIS

именах элементов. К тому же можно создавать группы идентификаторов с уникальными префиксами для своих областей (см. модуль *ARIS Identification* в разделе 5.12).

Идентификатор можно создать автоматически. Он состоит из префикса, который может быть легко определен, и числового индекса, задаваемого автоматически. В ARIS идентификатор может быть использован как альтернатива для имени объекта — во-первых, при создании копий объектов, не связанных между собой, и во-вторых, при объединении объектов, когда в качестве критерия для отбора совпадающих объектов и моделей могут служить как «имя», так и «идентификатор». Идентификатор может быть использован для фильтрации объектов (поиска, вывода в отчет). Следующие атрибуты заполняются автоматически:

- тип элемента (*Type*);
- время создания (*Time of generations*);
- автор (*Created by*) последнего изменения (*Last change*);
- последний пользователь (*Last user*).

Они выделены серым цветом и недоступны для изменения.

### 4.3. Методологические фильтры

Большое число моделей, включенных в ARIS, никогда не используется в рамках одного проекта. В этом просто нет необходимости, так как ни один объект не требует для описания такого числа моделей.

Для того, чтобы облегчить работу, обеспечив выбор ограниченного числа необходимых моделей, в ARIS реализованы *методологические фильтры*, представляющие собой регулируемые, переключаемые наборы моделей. Стандартные методологические фильтры ARIS приведены в табл. 1.

Таблица 1. Стандартные методологические фильтры ARIS

Название фильтра	Сокращенное название	Описание
Entire Method -полный фильтр	FULLMETHOD	Содержит все модели, объекты, отношения, символы и типы атрибутов, определенные в системе ARIS.
Easy Filter -простой фильтр	EASYFILTER	Содержит простейшие методы моделирования деятельности.
SAP Filter -SAP-фильтр	SAP-FILTER	Содержит методы моделирования, соответствующие моделям в SAP R/3.
Standard Filter -стандартный фильтр	STANDARD	Содержит наиболее часто используемые методы моделирования.
Extended Standard Filter - расширенный стандартный фильтр	EXTENDEDSTANDARD	В дополнение к стандартному фильтру содержит модели и типы объектов, которые предназначены для объектно-ориентированного моделирования.
ARIS for MySAP.com Фильтр для модуля ARIS mySAP.com	ARISMYSAPDOTCOM	Содержит методы моделирования, соответствующие моделям в SAP R/3 версии 4/6Б и сценариям e-бизнеса.
ARIS Simulation and ARIS ABC Filter - фильтр для модулей ARIS Simulation и ARIS ABC	SIMULATION	Содержит все модели, объекты, соединения, связи и атрибуты, которые применяются в ARIS Simulation и ARIS ABC.
Balanced Scorecard Filter-фильтр для модуля ARIS BSC	BALANCED SCORECARD	Содержит дополнительные модели для метода Balanced Scorecard.

### Основные элементы реализации системы ARIS

Методологический фильтр выбирается (рис. 33) через команду меню *Option (Опции)*, закладка *Log In (Подключение)*, кнопка *Browse (Просмотр)*. Пользователь может создавать свои методологические фильтры, для чего предусмотрен специальный механизм, называемый *Configuration Filter Wizard (Помощник конфигурирования фильтров)*. Он запускается командой *New (или Open)* контекстного меню, вызываемого на закладке *Filters (Фильтры)* базы данных *Configuration (Конфигурация)*.

В ARIS существует понятие стандартного методологического фильтра. *Стандартный методологический фильтр* — это фильтр, который присваивается базе. Он автоматически ставится в соответствие каждому новому пользователю и каждой новой их группе. По умолчанию, в качестве стандартного, каждой базе присваивается методологический фильтр *Easy Filter*.

Ответственным за управление фильтрами является модуль *ARIS Configuration*, рассмотренный в разделе 5.13.

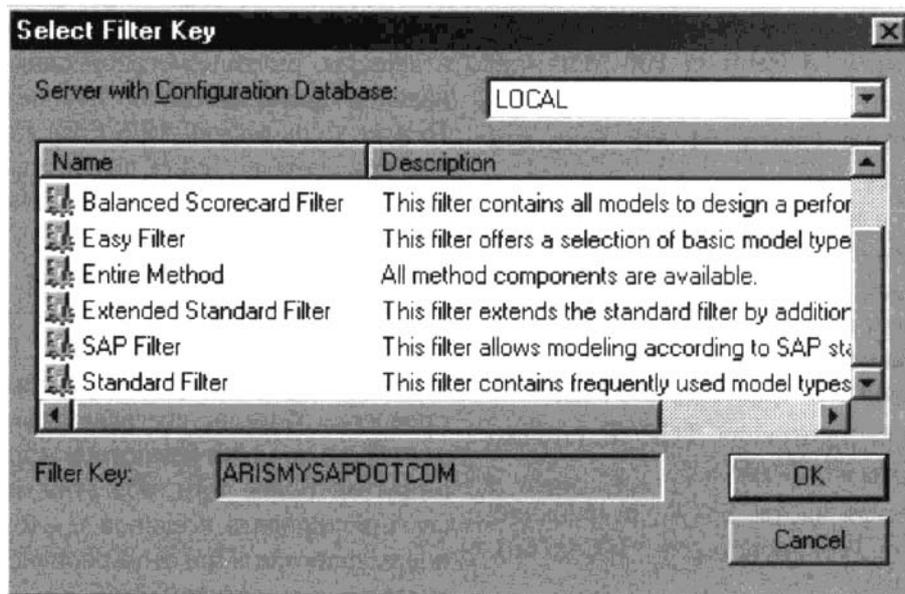


Рис. 33. Окно выбора методологического фильтра

## 5. Инструментальная система ARIS

### 5.1. Обзор основных модулей

Интегрированная инструментальная среда ARIS является сложной системой и состоит из комплекса взаимосвязанных и взаимодополняющих модулей, выполняющих различные функции. Их состав и взаимосвязи показаны на рис. 34.

Требования к компьютеру, на котором развертывается ARIS — довольно мягкие и сводятся к следующему:

- процессор Intel Pentium 166 MHz или более;
- оперативная память 64 MB (оптимум 128MB);
- свободное пространство жесткого диска — не менее 120 MB плюс объем, необходимый для баз данных пользователя. Последняя величина может быть весьма значительной, например, для базы данных модели SAP требуется 450 MB;
- поддержка протокола TCP/IP (только для работы в сети ARIS);
- графическая карта: SVGA, разрешение 640x480 (рекомендуется 1024x768), 256 цветов.

При использовании отдельных модулей требования к компьютеру могут повышаться. Так, для нормальной работы модуля ARIS Simulation необходимы 128 MB оперативной памяти и наличие протокола TCP/IP, для модуля ARIS Web Publisher в дополнение к сказанному желательна файловая система NTFS.

ARIS может работать под управлением операционных систем Window 95/98/2000 и Window NT 4.0 (Service Pack 4, MS Y2K Patch, Service Pack 5 или Service Pack 6a). Сетевая версия ARIS требует наличия сети Novell NetWare версий от 4.11 до 5.0.

Некоторые модули требуют специализированного программного обеспечения (рис. 34).

Основным модулем анализа и проектирования является базовый модуль *ARIS Toolset*, который может включать все остальные модули системы. При необходимости они легко интегрируются в *ARIS Toolset*.

Модуль *ARIS Easy Design* представляет собой упрощенное средство моделирования и анализа, имеющее ограниченные функциональные возможности по сравнению с *ARIS Toolset*. Если модуль *ARIS Easy Design* отсутствует в системе, то все подключенные к нему модули соединяются непосредственно с *ARIS Toolset*.

С точки зрения методологии, *ARIS Easy Design* поддерживает все методы моделирования, применяемые в *ARIS Toolset*. В этом отношении между ними существует полная совместимость. Модели, создаваемые в одном из этих средств, могут без потерь использоваться в другом. Более того, оба этих модуля могут функционировать в единой среде коллективной разработки. При этом рабочие места с установленным модулем *ARIS Easy Design* используются в соответствующих подразделениях организации для сбора информации и проведения общего анализа, а модуль *ARIS Toolset* служит средством для выявления более глубоких противоречий и оптимизации разработанных моделей.

Набор инструментов и функциональные возможности модулей *ARIS Easy Design* и *ARIS Toolset* определяют различия в областях их применения. *ARIS Easy Design* обычно используется оперативными аналитическими отделами организаций для проведения технического анализа, а инструменты *ARIS Toolset* более предназначены для управления бизнес-процессами. Следует отметить, что *ARIS Toolset* также включает в себя *ARIS Easy Design*.

## Инструментальная система ARIS

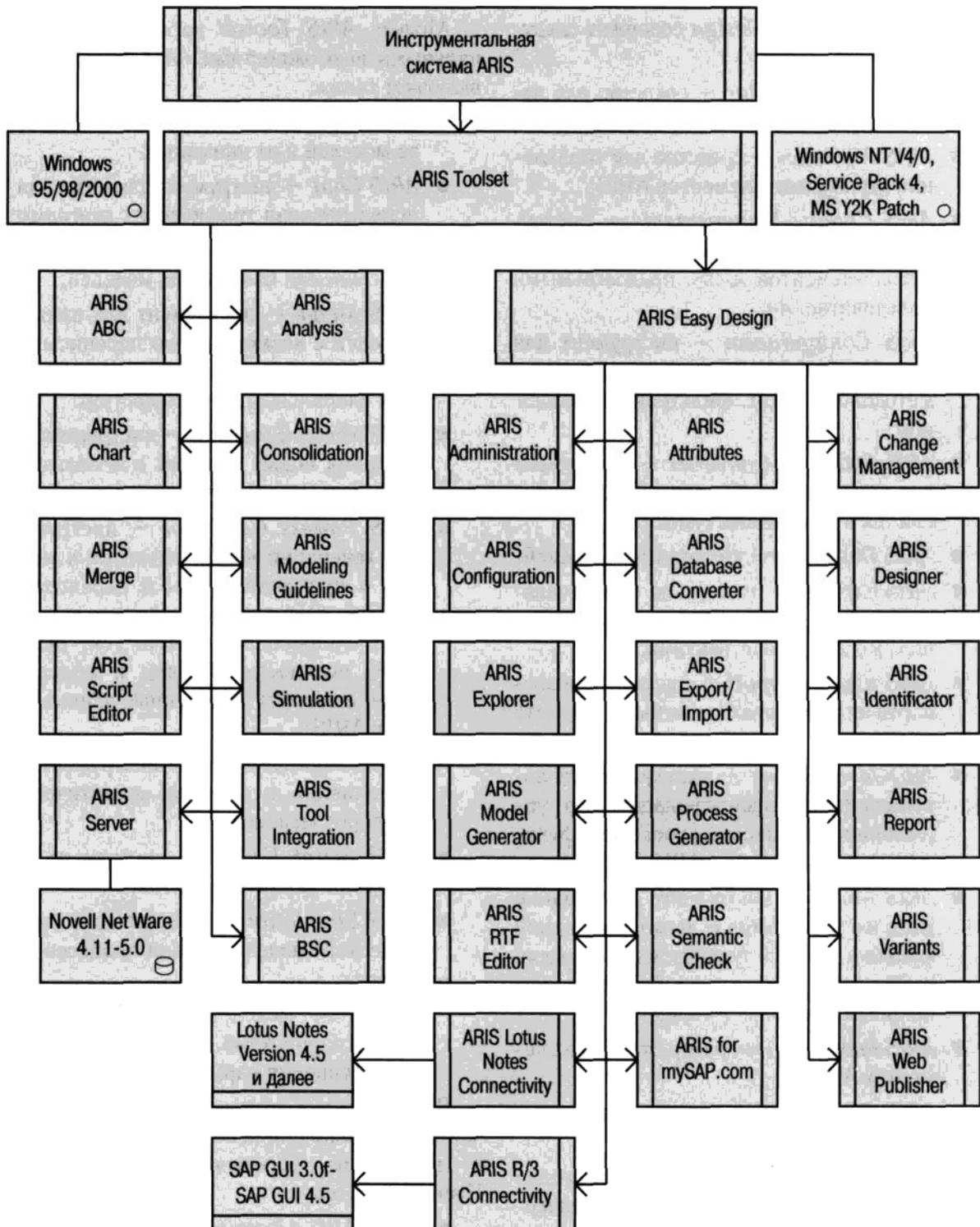


Рис. 34. Структура ARIS (системная организация), представленная с помощью диаграммы типа прикладной системы

### Моделирование бизнеса. Методология ARIS

Модуль *ARIS Easy Design* содержит следующие компоненты:

- *ARIS Administrator* — средство для администрирования баз данных ARIS;
- *ARIS Attributes* — средство для управления атрибутами элементов ARIS;
- *ARIS Change Managements* — инструмент для контроля за изменениями атрибутов элементов ARIS, предлагаемыми пользователями;
- *ARIS Configuration* — инструмент для импорта из одной базы данных в другую методологических фильтров и шаблонов;
- *ARIS Database Converter* — инструмент для конвертации баз данных ARIS версии 3-х в базы данных ARIS 5.0;
- *ARIS Designer* — конструктор моделей;
- *ARIS Explorer* — проводник, обеспечивающий работу с серверами, базами данных, моделями и объектами;
- *ARIS Export/Import* — средство для экспорта или импорта баз данных в формате ASCII;
- *ARIS Identifier* — инструмент для автоматической идентификации без управления префиксами (последняя функция встроена только в ARIS Toolset);
- *ARIS Merge* — инструмент для слияния моделей и объектов (с помощью данной функции нельзя переписывать модели и объекты с одинаковым GUID — глобальным идентификатором);
- *ARIS Report* — генератор отчетов о элементах ARIS;
- *ARIS RTF Editor* — редактор текстовых документов;
- *ARIS Semantic Check* — инструмент для выполнения (но не задания) семантических проверок моделей;
- *ARIS Variants* — средство для отображения вариантов моделей.

Модуль *ARIS Toolset* наряду с функциональными возможностями *ARIS Easy Design* включает также:

- *ARIS Analysis* — инструмент для анализа моделей и их анимации;
  - *ARIS Chat* — инструмент для создания и использования графических диаграмм;
  - *ARIS Consolidation* — инструмент для объединения баз данных моделей;
  - *ARIS Merge* — инструмент для слияния моделей с возможностью переписывать модели и объекты, имеющие одинаковый глобальный идентификатор.
  - *ARIS Model Generator* — инструмент для создания новых моделей с использованием уже существующих;
  - *ARIS Process Generator* — инструмент для генерации новых объектов и моделей в программе Excel и переноса их в ARIS. Имеется возможность с помощью специального отчета (см. раздел 10.3) перенести объекты и модели в Excel и вернуть измененные данные назад в ARIS;
  - *ARIS Semantic Check* — инструмент для выполнения и задания семантических проверок моделей;
  - *ARIS Script Editor* — инструмент для создания скриптов отчетов;
  - *ARIS Simulation* — средство для динамического моделирования процессов;
  - *ARIS Variants* — средство для создания вариантов моделей, удаления связей между вариантами, выполнения операций сравнения вариантов.
- В ARIS предусмотрены модули, предназначенные для решения некоторых частных задач и требующие дополнительных лицензий. К ним относятся:
- *ARIS ABC (Activity Based Costing)* — инструмент для проведения функционально-стоимостного анализа моделей;

### Инструментальная система ARIS

- *ARIS Connectivity for Lotus Notes* — инструмент, позволяющий запускать базы и документы Lotus Notes, которые связаны с элементами ARIS. Предусмотрена возможность открывать базы Lotus Notes, документы Lotus Notes, которые содержат элементы ARIS, с помощью веб-браузеров;

- *ARIS Connectivity for R/3* — инструмент для переноса моделей ARIS в формат HTML, использующий функциональные возможности транзакций информационной системы SAP R/3. Это позволяет запускать функции, связанные с системными операциями SAP R/3, из веб-браузеров;

- *ARIS Tool Integration* — инструмент для обмена информацией баз данных моделей ARIS с программными приложениями других производителей (с так называемыми приложениями партнеров);

- *ARIS Web Publisher* — инструмент для преобразования моделей ARIS в формат HTML с целью их использования как внутри компании, так и за ее пределами после передачи через Интранет или Интернет. Различные типы экспорта графики позволяют использовать модели ARIS на множестве платформ;

- *ARIS BSC (Balanced Scorecards)* — инструмент для стратегического управления.

Средства ARIS могут применяться как однопользовательская среда, а также поддерживать коллективные разработки в среде «клиент-сервер». Возможность коллективной работы обеспечивают:

- объединение баз данных (модуль *ARIS Merge*);

- обмен моделями через Интранет и Интернет (модуль *ARIS Web Publisher*);

- совместный доступ нескольких пользователей к единому хранилищу данных — репозиторию ARIS (модуль *ARIS Server*).

В состав ARIS входят вспомогательные специализированные модули, например, модуль *ARIS Script Converter*, предназначенный для конвертации скриптов, созданных в ARIS 4-х, в скрипты ARIS 5.0, модуль *ARIS Admintool*, предназначенный для управления базами данных ARIS при работе под Window NT и Novell, *ARIS for INTERSHOP infinity* и некоторые другие.

Один из дополнительных модулей — библиотека функций *ARIS API (Application Programming Interface)*. Он является интерфейсом между архивом данных ARIS и пользовательским приложением. Интерфейс выполнен в виде динамической библиотеки, интегрируемой в модуль *ARIS Toolset* по отдельной лицензии. Библиотека *ARIS API* содержит необходимые описания функций интерфейса, делающие возможным их использование в языках программирования C и C++.

Модуль *ARIS API* предоставляет доступ к данным, созданным с помощью системы ARIS, так что становится возможным использовать эту информацию в других приложениях или дополнять существующие данные в архиве ARIS новыми объектами.

При помощи библиотеки *ARIS API* могут быть решены следующие задачи:

- перенос из архива или в архив результатов моделирования в системе ARIS;

- создание своих надстроек к системе ARIS;

- интеграция информации в едином стандарте ARIS.

Существующая библиотека для языка C описывает функции интерфейса, позволяющие читать и вносить любые данные в архив ARIS. Работа функций с информацией производится с учетом прав доступа каждого пользователя. Разработчик библиотеки функций гарантирует, что изменения, вносимые с помощью функций, не разрушат архив ARIS, который остается доступным через средства модуля *ARIS Toolset*.

## Моделирование бизнеса. Методология ARIS

Библиотека API функций была разработана как динамически подключаемая библиотека, чтобы исключить зависимость от специфики конкретного компилятора языка программирования. Поэтому разработчик может выбрать удобную для себя программную среду.

Каждый модуль ARIS имеет свой логотип, изображаемый в правом верхнем углу основного окна.

Благодаря модульности комплекс средств ARIS позволяет скомпоновать оптимальный состав системы, полностью обеспечивающий реализацию необходимых задач, не оплачивая ненужные для конкретного применения возможности.

Некоторые модули ARIS рассмотрены далее более подробно.

## 5.2. Проводник — ARIS Explorer

### 5.2.1. Основные элементы управления

*ARIS Explorer* — это основное средство навигации в ARIS, предназначенное для управления базами данных, папками, моделями, объектами и другими элементами. Его построение (рис. 35) схоже с построением Проводника Microsoft Windows.

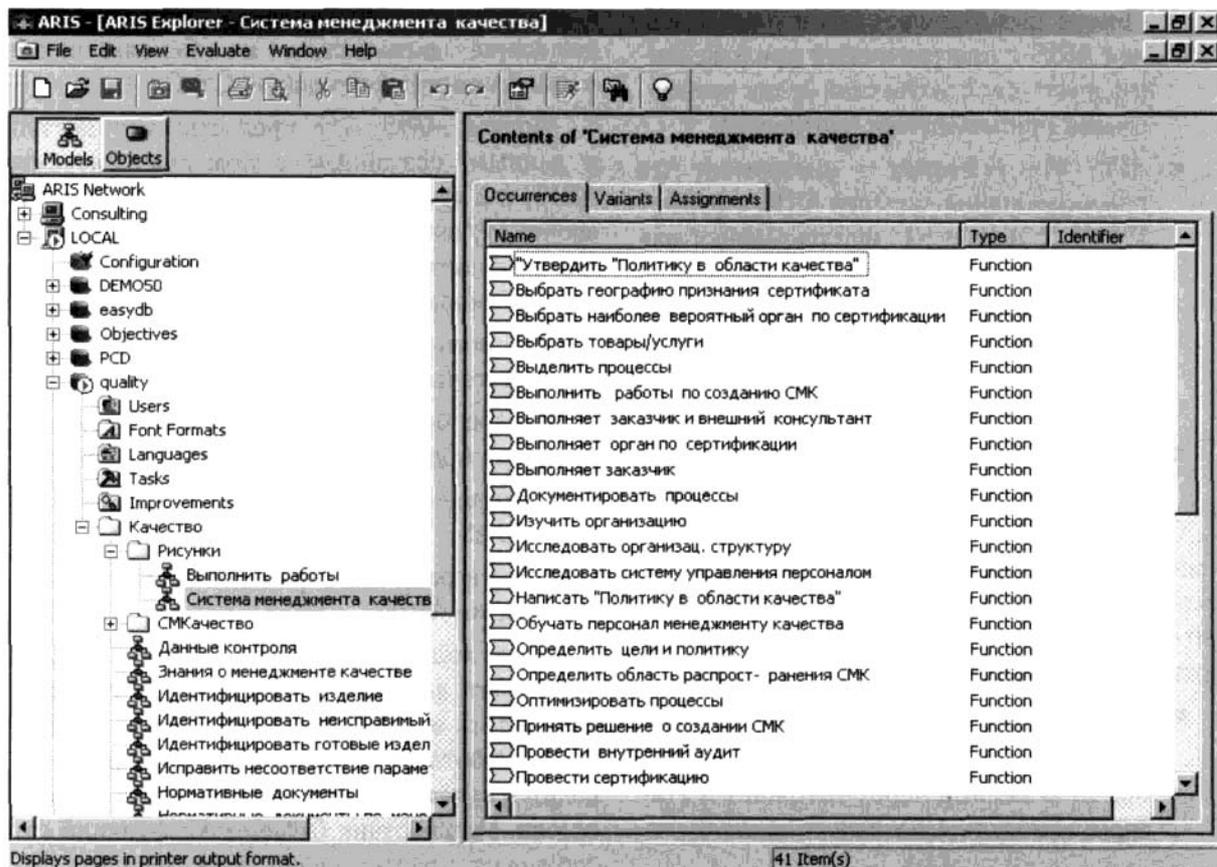


Рис. 35. ARIS Explorer

## Инструментальная система ARIS

К основным элементам, которые управляются посредством *ARIS Explorer* (Проводника), относятся:

- серверы;
- базы данных;
- пользователи и группы пользователей;
- пароли, определяющие различные привилегии (уровни) доступа к моделям;
- форматы шрифтов;
- языки баз данных;
- методологические фильтры;
- модели и содержащие их папки;
- объекты;
- таблицы.

В окне *ARIS Explorer* (рис. 35) отображаются все серверы баз данных, к которым имеется доступ. Используя проводник, можно определить новые соединения или удалить уже существующие.

Для запуска *ARIS Explorer* необходимо выбрать пункт меню *View/ARIS Explorer*. Для того, чтобы окно *Проводника* появлялось автоматически сразу после запуска *ARIS*, следует выбрать пункт меню *View /Options... (Опции)*, в появившемся на экране окне настроек (рис. 36) выбрать закладку *Explorer* и включить опцию *Start ARIS Explorer Automatically (Автоматический запуск Проводника)*. После этого на экране появится *oto ARIS Explogeg* (рис. 35).

Если пиктограммы *User* (Пользователи), *Font Format* (Настройки шрифтов), *Languages* (Языки), *Tasks* (Задачи) или *Improvements* (Изменения) не отображаются на экране, то необходимо выбрать пункт меню *View/Options*. В закладке *Explorer* следует включить нужные опции в группе настроек *Display* (Экран).

Окно *ARIS Explorer* (рис. 35) разделено на две части. В левой части в виде иерархической структуры, т.е. в форме дерева, представлены элементы *ARIS*. Слева от каждого из них находится пиктограмма, отображающая его тип. В корне дерева размещается элемент *ARIS Network (Сетевое окружение)*. Под ним располагаются серверы, папки и модели. (Подробно структура рассмотрена в разделе 4.)

После того, как в левой части окна выбран элемент структуры, в правой части окна появится список вложенных в него компонентов. Так, на рис. 35 в левой части окна показано содержимое папки *Терминология*.

Слева от компонента иерархической структуры может располагаться кнопка структурирования в виде квадрата со знаком «плюс» или «минус». Ее наличие говорит о том, что у данного компонента есть вложенные элементы. Развертывание/свертывание структуры вложенности можно получить любым из перечисленных ниже способов:

- дважды щелкнуть левой кнопкой мыши на нужном элементе структуры;
- выделить элемент структуры и нажать клавишу «+» или «—» на клавиатуре;
- щелкнуть левой кнопкой мыши на кнопке структурирования слева от нужного элемента структуры.

Если элемент из левой части окна *ARIS Explorer* не содержит вложенных структур, то его компоненты будут отражены в списке правой части, а действия по развертыванию/свертыванию вложенных структур не будут доступны для этого элемента.

В правой части окна отображается список элементов, подчиненных компоненту, выбранному в левой его части.

Меню *ARIS Explorer* расположено всегда на самом верху экрана. Состав команд может меняться в зависимости от активного пользовательского окна. Пункты меню могут быть доступными и недоступными, это зависит от текущего объекта в пользовательском окне.

Большинство пунктов меню и кнопок панелей инструментов *ARIS* являются традиционными для программных средств, работающих под *Windows*. Поэтому далее будут рассмотрены команды и действия, присущие только *ARIS*.

*File/Run (Выполнить)*. Выполнить программу, скрипт или ссылку для выбранного элемента структуры в окне *ARIS Explorer*. При этом выделенный элемент должен быть описан в атрибутах как тип *Link1*, *Link2*, *LinkS* или *Link4*.

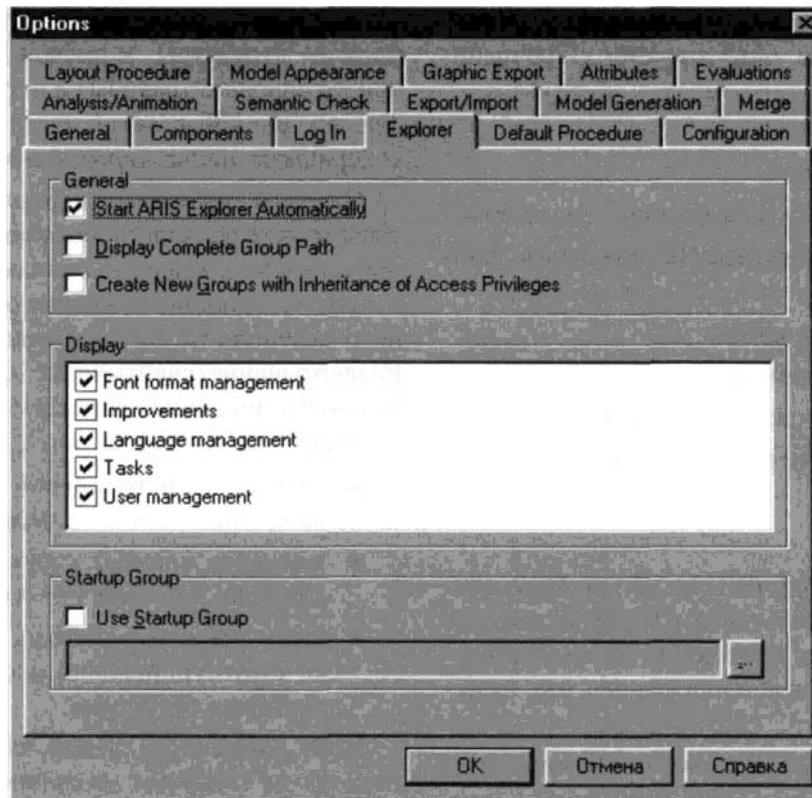


Рис. 36. Окно настроек ARIS Explorer

*Edit/Attributes (Атрибуты)*. Просмотр и редактирование атрибутов выделенных объектов. После выбора этого пункта меню на экране появится окно с описанием атрибутов объектов (рис. 53). Ответственным за выполнение этих действий является модуль *ARIS Attributes*, который рассмотрен в разделе 5.4.

*Edit/Find (Найти)*. В этом меню содержатся команды поиска моделей и объектов (пункт меню *Models and Objects* (Модели и объекты)). При выполнении команды на экран вызывается окно задания условий поиска (рис. 37). Поиск осуществляется в открытой базе данных. Если такой нет, то сначала следует открыть базу. В окне поиска указывается, что необходимо найти: модель или объект, задается тип модели или объекта, может быть указано конкретное имя, путь, по которому нужно искать, и т.д.

*Edit/Properties (Свойства)*. Данный пункт открывает окно *Options* (Свойства) (рис. 36), позволяющее настроить установленные элементы ARIS.

*View/Refresh (Обновить)*. Выполнение этого пункта меню обновляет информацию в окне *ARIS Explorer*.

*Evaluate/Report (Отчет)*. Данный пункт обеспечивает получение отчета для выбранных элементов структуры. После выбора этого пункта меню система последовательно предложит несколько диалоговых окон, в которых следует указать параметры отчета. Подробнее это описано в разделе 10.

*Evaluate/Analysis (Анализ)*. После выбора этого пункта на экране появится последовательность окон с инструкциями, выполнение которых поможет настроить параметры работы и получения аналитической информации. Подробнее данный вопрос изложен в разделе 9.7.

## Инструментальная система ARIS

*Evaluate/Semantic Check* (Семантическая проверка). Выбор этого пункта приводит к появлению на экране окна с настройками пара-

метров семантической проверки выбранной модели. Семантические проверки, обсуждаются в разделе 9.6.

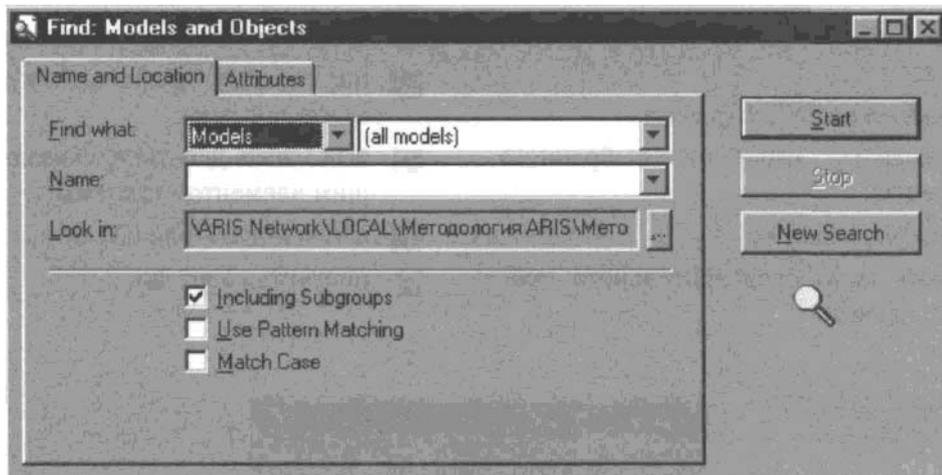


Рис. 37. Окно поиска моделей и объектов

Для упрощения работы в системе ARIS предусмотрены различные панели инструментов. В зависимости от того, какой модуль ARIS используется в данный момент, доступны те или иные панели инструментов. Для их подключения и/или отключения следует произвести любое из перечисленных действий:

- выполнить команду меню *View/ Toolbars* (Вид/Панели инструментов)^ в выпадающем списке подключить/отключить нужную панель;

- подвести курсор мыши к области расположения панели инструментов (вверху основного окна системы, под командами меню) и щелкнуть правой кнопкой мыши, затем в выпадающем контекстном меню подключить/отключить панель.

Во время работы в окне *ARIS Explorer* могут понадобиться перечисленные ниже панели инструментов.

*File* (Файл). На этой панели сосредоточены команды для работы с элементами баз данных (рис.38)



Рис. 38. Панель инструментов File

Кнопки, расположенные на этой панели, позволяют:

-  создать новый элемент базы данных;
-  открыть существующий элемент базы данных;
-  сохранить изменения текущего элемента;
-  открыть окно *ARIS Explorer*;
-  просмотреть атрибуты выбранного элемента;
-  напечатать модель;
-  открыть окно предварительного просмотра печати;

-  вырезать выделенные элементы;
-  копировать выделенные элементы;
-  вставить предварительно вырезанные или скопированные элементы;
-  отменить последнее действие;
-  повторить последнее действие;
-  просмотреть свойства текущего элемента системы;
-  выполнить действие, связанное с текущим элементом системы;
-  найти модель или объект;
-  получить совет дня.



Рис. 39. Панель инструментов Explorer

*Explorer (Проводник)*. На этой панели (рис. 39) сосредоточены команды для управления элементами, находящимися в правой части окна *ARIS Explorer*.

Кнопки на панели *Explorer* представляют возможность:

-  перейти на один уровень вверх в структуре проводника;
-  отобразить элементы правой части окна в виде больших иконок;

-  отобразить элементы правой части окна в виде маленьких иконок;
-  отобразить элементы правой части окна в виде списка;
-  отобразить элементы правой части окна в виде списка с подробным описанием;
-  отсортировать по имени;
-  отсортировать по типу;
-  удалить текущий элемент структуры.



Рис. 40. Панель инструментов ARIS Explorer

*ARIS Explorer (Проводник ARIS)*. На этой панели сосредоточены команды для создания новых элементов системы (рис. 40).

Кнопки на панели *ARIS Explorer* предназначены для выполнения команд:

-  создать новую папку;
-  создать новый объект.

Существуют также другие панели инструментов:

- Animation Control (Управление анимацией);
- Attributes (Атрибуты);
- Chart (Диаграммы);
- Debug (Отладчик);
- Insert (Вставка);
- Formatting (Формат);
- Macros (Макросы);
- Model View (Вид моделей);
- Modeling (Моделирование);
- Script Layout (Макет скриптов);
- RTF Editor (Текстовый редактор).

Эти панели рассматриваются далее в соответствующих разделах книги. Следует отметить, что каждая кнопка на панелях снабжена всплывающей подсказкой, что значительно упрощает использование панелей инструментов.

В дополнение к командам основного меню ARIS для каждого элемента структуры определено *контекстное меню*, которое позволяет ускорить процесс редактирования элементов. Для вызова контекстного меню необходимо:

- выделить нужный элемент;
- щелкнуть правой кнопкой мыши на этом элементе.

Для того, чтобы редактировать несколько элементов одновременно, необходимо:

- выделить несколько элементов (в правой части окна *ARIS Explorer*). Для этого следует нажать на клавиатуре клавишу *Ctrl* и, не отпуская ее, выделить курсором мыши нужные элементы;

- переместить курсор мыши за пределы выделения и щелкнуть правой кнопкой мыши. После этого на экране появится контекстное меню для выбранного элемента (или группы элементов) структуры. Набор команд контекстного меню изменяется в зависимости от места его вызова.

Следует обратить внимание на то, что выделение группы объектов возможно только в правой части окна *ARIS Explorer*.

Во всплывающем меню есть команды, которые не могут быть выполнены для группы объектов, например, *Rename (Переименовать)*. В этом случае, независимо от выделенных элементов, эта команда будет выполнена для элемента, находящегося в текущий момент под курсором мыши.

## 5.2.2. Компоненты сетевого окружения ARIS — ARIS Network

*Сетевое окружение ARIS* — это корневой элемент в структуре данных ARIS. Он включает в себя все остальные элементы системы. Структура сетевого окружения ARIS подробно рассмотрена в разделе 4.1.

Действия, которые пользователь может производить с компонентами сетевого окружения, определяются активными пунктами его контекстного меню. Используя команды меню, можно выполнить следующие действия:

*Open (Открыть)* — развернуть вложенные структуры для элемента *ARIS Network*. Вложенными для данного элемента являются серверы баз данных, которые включены в структуру системы;

*Explore (Просмотреть)* — развернуть структуру Проводника для элемента *ARIS Network*.

Следующими после *ARIS Network* в структуре компонентов системы являются серверы баз данных. Для того, чтобы просмотреть структуру сервера баз данных, нужно в левой части окна выделить курсором элемент *ARIS Network*, и тогда в правой части окна будет представлен список всех подключенных серверов. Можно также воспользоваться командами контекстного меню, которое содержит команды для работы с сервером баз данных:

*Open (Открыть)* — развернуть структуру для текущего сервера баз данных. Вложенными для данного элемента являются базы данных;

*Explore (Просмотреть)* — развернуть структуру *ARIS Explorer* для текущего элемента.

При позиционировании курсора на конкретном сервере баз данных система автоматически произведет соединение с этим сервером, если к этому моменту оно еще не было установлено. При успешном соединении в пиктограмме справа от имени сервера появится стрелка в кружочке. В противном случае система выдаст на экран сообщение о невозможности соединения, и на пиктограмме появится красный кружок.

Команда контекстного меню *Disconnect* (*Прервать соединение*) позволяет прервать соединение с текущим сервером баз данных. После выбора этого пункта система отключит сервер, и он пропадет из окна *ARIS Explorer*,

Для подключения нового сервера баз данных необходимо вызвать контекстное меню на элементе *ARIS Network* и выполнить команду *Connect Server* (*Соединение с сервером*). Она позволяет добавить в структуру новый

сервер баз данных ARIS для дальнейшего подключения к нему. После выбора этого пункта на экране откроется диалоговое окно *New Server* (*Новый сервер*) (рис. 41).

В поле *Server Name or IP Address* (*Имя сервера или IP-адрес*) необходимо указать имя сервера или его IP-адрес. Для этого можно воспользоваться кнопкой выбора справа от поля. Если включить опцию *Check if Server is available* (*Проверить, доступен ли сервер*), то непосредственно перед добавлением сервера система проверит, доступен он или нет. Если нет, то будет предложено либо указать другой сервер, либо вообще отказаться от добавления нового сервера. После того, как вся необходимая информация в диалоговом окне будет указана, нужно выбрать кнопку *OK* для добавления в структуру нового сервера. Для отказа нажимают традиционную кнопку *Cancel*

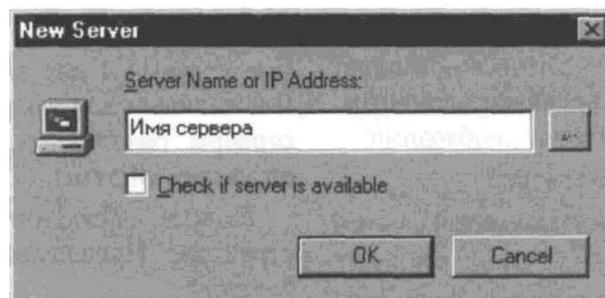


Рис. 41. Диалоговое окно подключения к новому серверу

Сервер баз данных включает *базы данных*, созданные пользователем. Базы данных содержат в себе настройки пользователей для фонтов, языков, а также набор моделей и объектов (подробнее см. раздел 4.1).

Для того, чтобы получить информацию о базах данных, расположенных на конкретном сервере, необходимо выполнить одно из следующих действий:

- в левой части окна выделить курсором нужный сервер, и тогда в правой части окна будет представлен список всех баз данных этого сервера;
- раскрыть структуру вложенности для нужного сервера, щелкнув на нем дважды левой кнопкой мыши.

### Инструментальная система ARIS

Можно также воспользоваться командами контекстного меню:

*Explore (Просмотреть)* — развернуть окно модуля *ARIS Explorer* для текущего элемента;

*Open (Открыть)* — открыть текущую базу данных. Базы, созданные с помощью более ранних версий системы, должны быть регенерированы перед их открытием. Перед открытием такой базы система выдаст на эк-

ран окно с запросом на регенерацию. Выберите кнопку *Yes (Да)*, после этого начнется регенерация, и на экран будут выводиться сообщения о ее шагах.

Для того, чтобы *открыть базу*, нужно подсоединиться к ней как зарегистрированный пользователь с необходимым методологическим фильтром. Существует два способа соединения с базой.



Рис. 42. Окно Login Wizard (Помощник соединения). Запрос имени и пароля пользователя

Если используются установки по умолчанию (пункт меню View/Options (Вид/Опции) и закладка Log In (Осуществить соединение), включена опция Use Defaults (Использовать установки по умолчанию) и помечено поле Filter Defaults (Установка фильтра по умолчанию)), то система будет пытаться осуществить соединения с этими параметрами каждый раз, когда курсор позиционируется на базе в левой части окна или дважды нажата левая кнопка мыши на базе.

Если настройки по умолчанию не используются, то каждый раз при выполнении соединения с базой на экране будет появляться окно

Login Wizard (Помощник соединения), где необходимо заполнить требующиеся поля ввода данных (рис. 42).

В окне в полях User (Пользователь) и Password (Пароль) необходимо указать имя и пароль пользователя. После выбора кнопки Далее на экране появится следующее окно с запросом фильтра соединения (рис. 43). После выбора нужного фильтра нажимается кнопка Готово.

При позиционировании курсора на конкретной базе данных система автоматически произведет соединение с этой базой, если к этому моменту оно еще не было установлено.

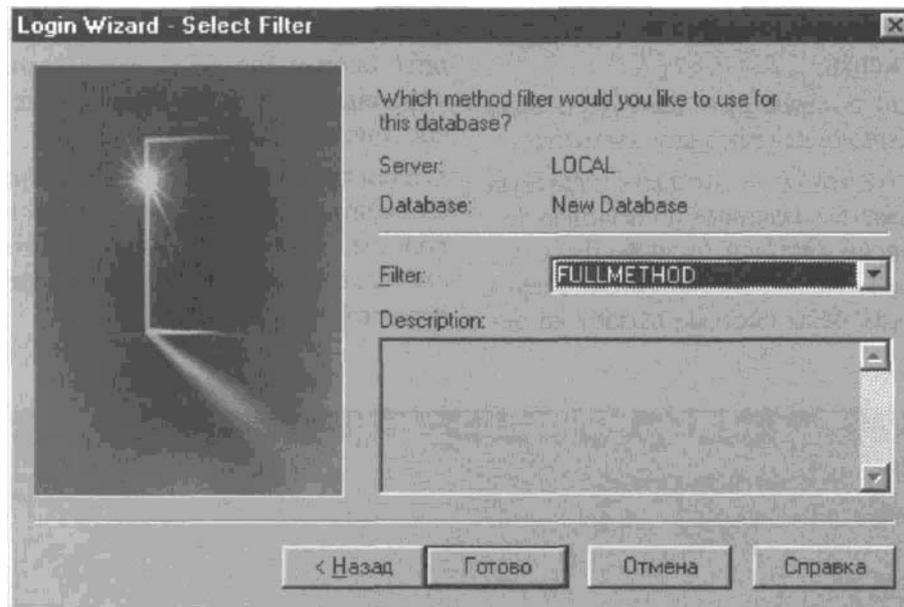


Рис. 43 Окно Login Wizard. Запрос методологического фильтра

При успешном соединении в пиктограмме справа от имени базы появится зеленая стрелка, в противном случае система выдаст на экран сообщение о невозможности соединения, и на пиктограмме появится красный кружок.

Команда контекстного меню *Log In* (*Осуществить соединение*) позволяет установить связь с текущей базой. На экране появится окно *Login Wizard* для ввода требуемых параметров (рис. 42 - 43).

Команда контекстного меню *Log Out* (*Отсоединиться*) позволяет разорвать связь с базой данных. После выполнения данного пункта меню система отсоединится от базы, и на пиктограмме справа пропадет стрелка.

Команда контекстного меню *Rename* (*Переименовать*) позволяет переименовать базу данных. Переименовать базу данных может только Администратор сервера и только в том случае, если к этой базе никто из пользователей не подсоединен. Если конкретный пользователь не был подсоединен к серверу как Администратор, или если в настройках подключения (меню *View /Options*, закладка *Login* (*Осуществить соединение*)) не включена опция *Use Defaults* (*Использовать установки по*

*умолчанию*), или если указанные параметры подключения некорректны, то на экран будет выдано окно с запросом пароля.

Для переименования можно также воспользоваться командой меню *Edit/Rename* (*Переименовать*) или клавишей *F2*. Имя базы должно содержать только буквы, цифры, пробел, дефис или символ подчеркивания. Нельзя использовать символ перевода строки. Имя базы должно быть уникально, и если база с таким именем уже существует, то система выдаст на экран предупреждение и предложит ввести новое имя.

Следующим компонентом сетевого окружения ARIS являются *пользователи* (*Users*) и *их группы*. Для того, чтобы получить информацию о зарегистрированных пользователях базы данных, необходимо в левой части окна выделить курсором соответствующий элемент *Users* (*Пользователи*), и тогда в правой части окна будет представлен список всех зарегистрированных пользователей текущей базы данных.

Для вызова контекстного меню необходимо щелкнуть правой кнопкой мыши на пиктограмме *Users*, и на экране появится меню, содержащее следующие команды

*Open (Открыть)* — открыть список пользователей в правой части окна *ARIS Explorer*;

*Explore (Просмотреть)* — развернуть структуру проводника для текущего элемента.

Функции для работы с пользователями системы доступны через контекстное меню в правой части окна *ARIS Explorer*. Используя их, можно создать нового пользователя или группу пользователей, изменить пароль пользователя, назначить привилегии доступа к функциональным возможностям системы, указать доступные фильтры и т.д. Подробнее эти вопросы рассмотрены в разделе 9.8.

*Настройки шрифтов (Font Formats)* — это стандартный компонент базы данных, который создается по умолчанию вместе с базой данных. Настройка шрифтов проводится при помощи команд контекстного меню, которые позволяют:

- создать новую настройку шрифта (команда *New/Font Format*);
- изменить существующую настройку свойств (команда *Properties*).

*Настройки языка (Languages)* — это стандартный компонент базы данных, который создается по умолчанию вместе с базой данных. Команды контекстного меню позволяют:

- создать новую настройку языка (команда *New/Language*);
- изменить существующую настройку свойств (команда *Properties*).

Объекты и модели, созданные пользователями системы, объединяются в *папки*. При работе с папкой доступны следующие команды всплывающего меню:

- *Open (Открыть)* — открыть (развернуть) структуру папки в правой части окна *ARIS Explorer*;
- *Explore (Просмотреть)* — развернуть структуру для текущей папки;
- *Rename (Переименовать)* — переименовать папку. Для переименования можно также воспользоваться командой меню *Edit/Rename* или клавишей *F2*;

- *Delete (Удалить)* — удалить выбранную папку. Перед удалением система выдаст на экран окно с запросом подтверждения удаления;

- *New (Новый)* — создать новый элемент папки;

- *Group (Папка)* — создать новую папку. После этого в списке правой части окна *ARIS Explorer* появится новая папка, которой будет присвоено по умолчанию имя (*New group*). Имя впоследствии можно изменить;

- *Object (Объект)* — создать новый объект. На экране появится диалоговое окно *Object Wizard (Помощник создания объекта)*, которое поможет создать объект;

- *Model (Модель)* — создать новую модель. На экране появится диалоговое окно *Model Wizard (Помощник создания модели)*. Необходимо выполнить приведенные в нем инструкции;

- *Run (Выполнить)* — запустить на выполнение программу, определенную для текущей папки.

Папка содержит в себе *модели* и *объекты* пользователей. Для того, чтобы получить информацию о существующих моделях, необходимо выполнить действия, описанные ниже.

В левой части окна выделяется папка, в которой находятся модели. При этом в правой части будет представлен список всех моделей (закладка *Models*) и объектов (закладка *Objects*) этой папки. Если же в левой части окна выделить курсором конкретную модель, то в правой его части появится список объектов (закладка *Occurrences (Содержание)* этой модели.

Для того, чтобы *создать новую модель*, необходимо выбрать папку, в которой она будет располагаться, и нажать пункт контекстного меню *New/Model...* или пункт меню *File/ New*. После этого в диалоговом окне необходимо выбрать тип нового документа *Model*. Контекстное меню содержит следующие команды:

## Моделирование бизнеса. Методология ARIS

- *Open (Открыть)* — открыть модель для редактирования. После выбора этой команды на экране появится окно *ARIS Designer*,
- *Cut (Вырезать)* — копировать текущую модель (с удалением из содержащей ее папки) для последующего переноса этой модели в другую папку;
- *Copy (Копировать)* — копировать текущую модель для последующего переноса копии в другое место;
- *Create Shortcut (Создать ярлык)* — создать ярлык для текущей модели;
- *Rename (Переименовать)* — переименовать элемент структуры. При вводе имени, содержащего несколько строк, целесообразно использовать комбинацию клавиш *Ctrl+Enter* для переноса строки. После ввода нового имени необходимо нажать клавишу *Enter*. Для переименования можно также воспользоваться командой меню *Edit/ Rename* или клавишей *F2*;
- *Delete (Удалить)* — удалить текущую модель. Перед удалением система попросит подтвердить это действие;
- *Print (Печать)* — распечатать модель на установленный по умолчанию принтер;
- *Run (Выполнить)* — запустить процесс выполнения данной модели.

Для редактирования модели можно воспользоваться командой *Open* всплывающего меню или дважды щелкнуть левой кнопкой мыши на нужной модели. После этого на экране появится окно *ARIS Designer*.

Модели состоят из *объектов*. Для того, чтобы получить информацию о существующих в папке объектах, необходимо в левой части окна выделить курсором соответствующую папку. Тогда в правой части окна появится список объектов (закладка *Objects*) этой папки. Если в левой части окна выделить конкретную модель, то список объектов этой модели (закладка *Occurrence*) отобразится справа.

Если дважды щелкнуть левой кнопкой мыши на объекте, то на экране появится окно с атрибутами или свойствами текущего объекта.

Для вызова контекстного меню нужно щелкнуть правой кнопкой мыши на объекте, после чего на экране появится меню, содержащее следующие команды:

- *Copy (Копировать)* — копировать текущий объект в буфер Windows для последующего переноса копии в другое место;
- *Rename (Переименовать)* — переименовать объект. Для ввода имени, содержащего несколько строк, используется комбинация клавиш *Ctrl+Enter*. Ввод нового имени завершается нажатием клавиши *Enter*. Для переименования можно также воспользоваться командой меню *Edit/ Rename* или клавишей *F2*.

Для того, чтобы найти конкретную модель или объекты, можно воспользоваться командой меню *Edit/Find* или командой *Find* контекстного меню.

## 5.3. Конструктор моделей — ARIS Designer

### 5.3.1. Основные элементы управления

*ARIS Designer* используется для создания и корректировки моделей. Построенные модели могут быть разделены на части, объединены, могут выступать в роли эталонных моделей, которые, в свою очередь, используются при создании новых моделей.

Меню *ARIS Designer* всегда расположено на самом верху экрана. В данном разделе описаны только те команды, которые являются специфичными для *ARIS Designer*.

*Edit/Find in active model (Найти в текущей модели)* — команда поиска объектов в текущей модели. Поиск ведется по типу объекта и по маске имени. Ввод параметров поиска осуществляется в диалоговом окне (рис. 44).

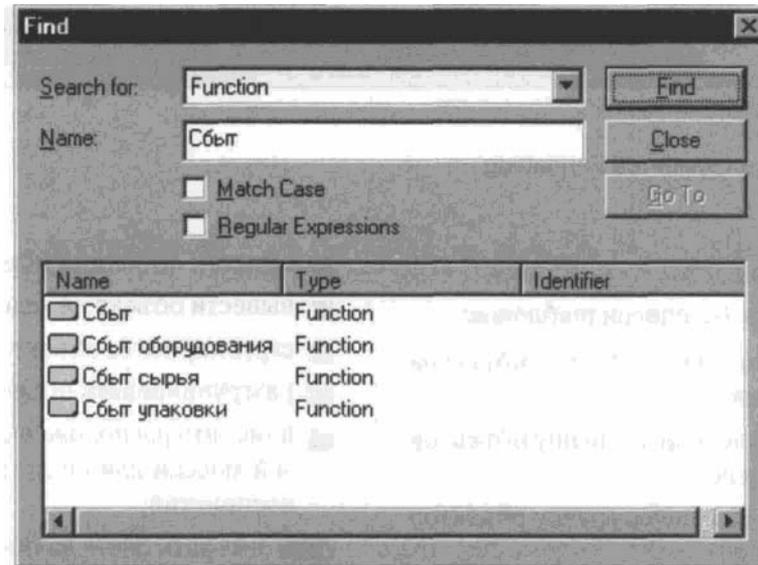


Рис. 44. Окно поиска объектов в текущей модели

Командная кнопка *Go To (Перейти)* обеспечивает переход к выбранному в списке объекту. Опции поиска традиционны.

Во время работы в окне *ARIS Designer* доступны следующие панели инструментов:

Model View (Просмотр модели).

На этой панели сосредоточены команды для управления просмотром модели (рис. 45).

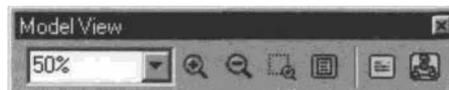


Рис. 45. Панель инструментов Model View

Приведенное здесь и далее в этом разделе описание кнопок слева направо:

- открыть выпадающий список с процентами масштабирования изображения модели;
- увеличить масштаб изображения;
- уменьшить масштаб изображения;
- выделить часть модели и увеличить масштаб ее просмотра;
- вернуться в нормальный (стандартный) режим просмотра;

- развернуть изображение модели на весь экран. Возврат в нормальный режим осуществляется нажатием клавиши *Esc*;
- выбрать оптимальный режим просмотра модели. Ее размер модели изменяется таким образом, чтобы вся она поместилась на экране.

**Formating (Форматирование).** На этой панели находятся команды управления размещением объектов модели (рис. 46).

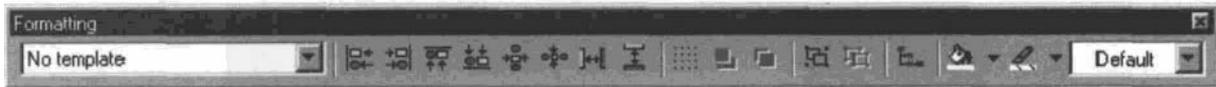


Рис. 46 Панель инструментов Formatting

Описание кнопок:

- открыть список шаблонов;
- выровнять выделенную группу объектов по левому краю;
- выровнять выделенную группу объектов по правому краю;
- выровнять выделенную группу объектов по краю сверху;
- выровнять выделенную группу объектов по краю снизу;
- выровнять выделенную группу объектов относительно центра по вертикали;
- выровнять выделенную группу объектов относительно центра по горизонтали;
- выровнять расстояния по вертикали между объектами группы;
- выровнять расстояния по горизонтали между объектами группы;
- показать/скрыть сетку;

- вывести объект на передний план;
- вывести объект на задний план;
- сгруппировать объекты;
- разгруппировать объекты;
- изменить расположение объектов и связей модели для ее лучшего визуального восприятия;
- выбрать цвета заливки;
- выбрать цвета линий;
- открыть список для выбора толщины линий.

**Modeling (Моделирование).** На этой панели сосредоточены кнопки, позволяющие устанавливать связи и выбирать типы объектов при моделировании. Конфигурация этой панели может отличаться в зависимости от типа текущей модели. Различные виды панели *Modeling* рассмотрены для каждой модели в разделах 6-8.



Рис. 47 Панель инструментов Insert

**Insert (Вставка).** На этой панели сосредоточены команды управления объектами модели (рис. 47).

Описание кнопок:

- вставить объект;
- вставить таблицу *Excel*;
- вставить документ текстового редактора *WordPad*;
- вставить прямоугольник;
- вставить скругленный прямоугольник;
- вставить овал;
- вставить круг;
- вставить точку;
- вставить произвольный объект из логической линии;
- вставить текстовое поле (комментарий);
- вставить атрибут модели, который выбирается из раскрывающегося списка.

### 5.3.2. Моделирование

Процесс моделирования начинается с создания новой модели. Это можно сделать несколькими способами:

- выбрать пункт меню *File/New*,
- щелкнуть на кнопке *New* панели инструментов;

- нажать комбинацию клавиш *Ctrl+N*.

После каждой команды на экране появится окно со списком типов элементов системы. Нужно выбрать в списке тип *Model* и нажать кнопку *OK*. На экране появится окно выбора папки для размещения новой модели. После выбора папки и нажатия кнопки *Далее* откроется окно *Model Wizard (Помощник конструирования моделей)* (рис. 48).

Типы моделей делятся на группы в соответствии с типами представлений организации (см. раздел 3): *Organization* (Организационные модели), *Data* (Модели данных), *Processes* (Модели процессов), *Functions* (Функциональ

ные модели). Эти группы, в свою очередь, делятся на уровни описания: Определение требований (*Requirements Definition*), Спецификация проекта (*Design Specification*) и Описание реализации (*Implementation*).

Необходимо указать группу или несколько групп, к которым относится выбираемый тип модели. Это можно сделать при помощи команд контекстного меню, вызываемого щелчком правой кнопки мыши на элементе *ARIS House (Select All* — выбрать все элементы, *Levels* — включить/отключить уровни описания), или же щелкая левой кнопкой мыши на элементах *ARIS House*, при этом отобранные представления будут выделены желтым цветом. После выбора типа представления модели выбирается тип модели из списка *Model Type* (в этом списке представлены модели, удовлетворяющие условиям текущего фильтра). Затем нажимается кнопка *Next*, и на экране появляется следующее окно *Model Wizard* (рис. 49).

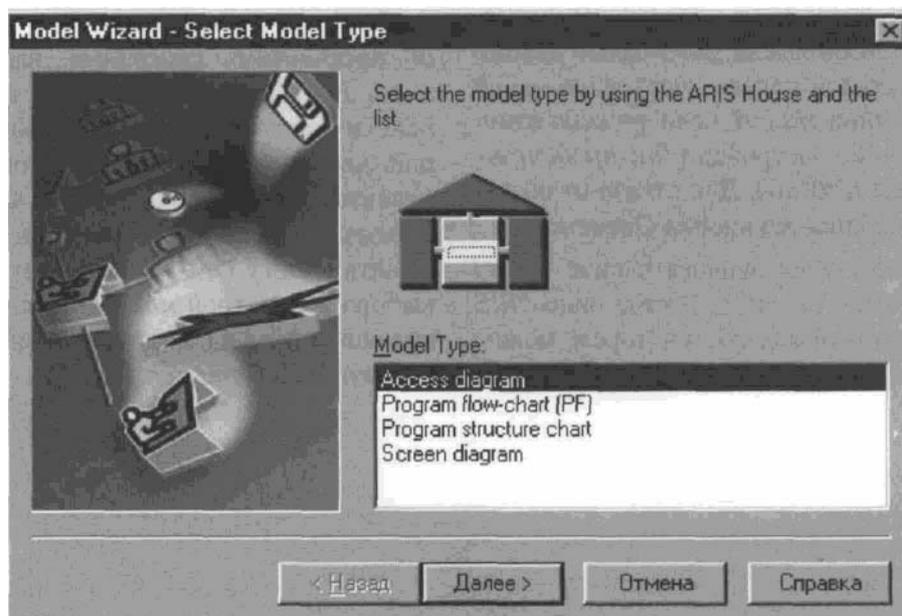


Рис. 48 Окно Model Wizard. Выбор типа новой модели

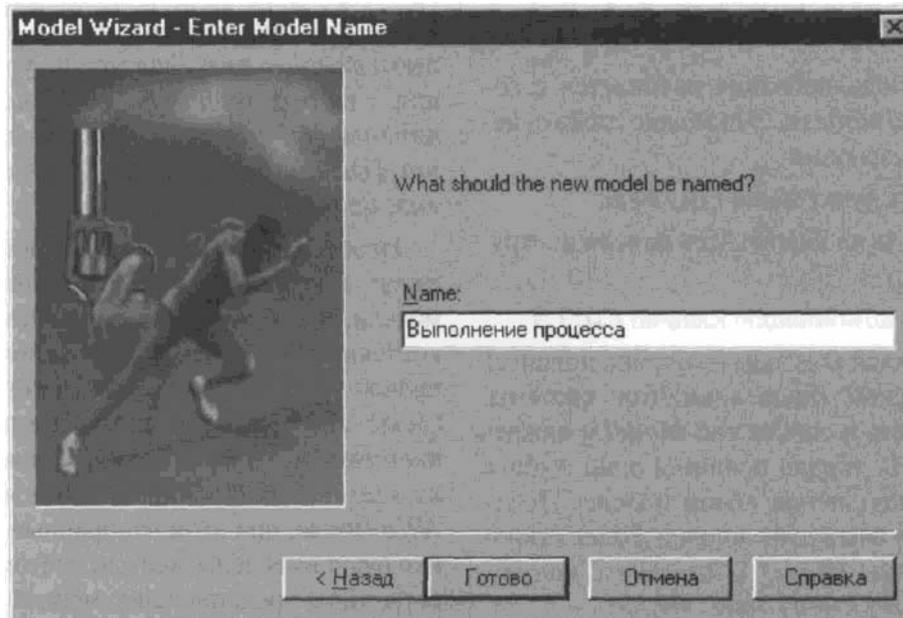


Рис. 49. Окно Model Wizard. Ввод имени новой модели

В поле *Name* указывается уникальное имя модели. В одной папке не может существовать двух моделей одного типа с одинаковыми именами. После ввода имени необходимо нажать кнопку *Готово* для начала построения новой модели или кнопку *Назад*, если решено изменить предыдущие настройки (*Тип представления* или *Тип модели*). Для отказа от создания модели выбирается кнопка *Отмена*.

После завершения диалога в окне *Model Wizard* на экране появится пустое окно *ARIS Designer* для новой модели, в котором можно создавать и изменять объекты и связи между

ними, а также вводить атрибуты объектов, связей и самой новой модели.

*Открыть существующую модель* можно несколькими способами: выбрать пункт меню *File/Open*, щелкнуть на кнопке *Open* панели инструментов или нажать комбинацию клавиш *Ctrl+O*. После этого на экране появится окно со списком типов элементов системы. В списке нужно выбрать тип *Model* и нажать кнопку *OK*. На экране появится окно выбора конкретной модели. После выделения нужной модели и нажатия кнопки *OK* откроется окно *ARIS Designer* (рис. 50).

## Инструментальная система ARIS

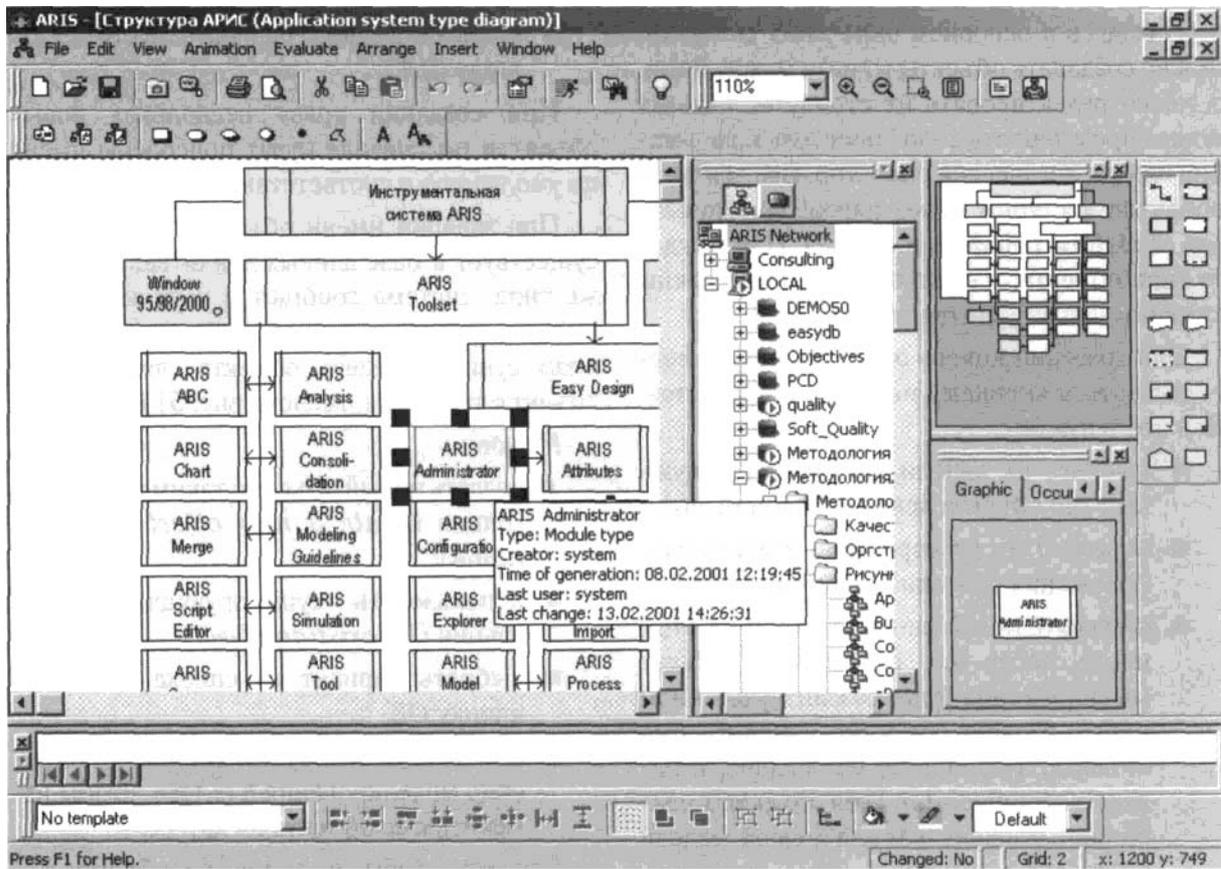


Рис. 50. Окно ARIS Designer

Окно *ARIS Designer* состоит из основного окна, где происходит построение моделей, и вспомогательных окон, которые открываются и закрываются командами меню. Используя команды пункта меню *View* или кнопки панели инструментов *Model View*, можно управлять форматом отображения информации в основном окне *ARIS Designer*

К основному окну моделирования может быть добавлено еще несколько ниже перечисленных вспомогательных окон. В скобках указаны команды меню, которые активизируют эти окна.

*Окно Просмотра модели* (команда меню *View/Model Overview*). В этом окне (на рис. 50 справа в верхней части) представлена модель в уменьшенном масштабе. Белый прямоугольник — это часть модели, которая отражается в основном окне *ARIS Designer*. Передвигая этот прямоугольник курсором мыши, можно изменять представление в основном окне.

*Окно просмотра объекта* (команда меню *View/Object Window*) — вспомогательное окно

для работы с выбранным объектом модели (он окружен маленькими черными квадратами на рис. 50). Оно используется для рассмотрения его в увеличенном масштабе, просмотра его связей и т.п. На рис. 50 это окно расположено справа в нижней части.

*Окно вывода информации* (команда меню *View/Output Window*) — вспомогательное окно для вывода системной информации для пользователя (внизу рис. 50).

Окно проводника (команда меню *View/Explorer Window*) — окно *ARIS Explorer* (правее центра на рис. 50).

### Моделирование бизнеса. Методология ARIS

Находясь в основном окне *ARIS Designer*, можно создавать объекты и связи между ними, а также редактировать их атрибуты. Объект может представлять собой носитель информации или же логический оператор. Все эти типы объектов доступны через панель инструментов *Modeling (Моделирование)*. Для каждого типа объекта, в свою очередь, определены символы представления его в модели.

Для создания нового объекта следует воспользоваться кнопками панели инструментов. Для этого нужно:

- выбрать на панели инструментов нужный тип объекта левой кнопкой мыши;
- переместить курсор мыши в место расположения объекта на модели;
- щелкнуть левой кнопкой мыши на этом месте, после чего графический объект заданного типа будет создан в указанном месте. Можно сразу же изменить имя объекта, заданное по умолчанию, и нажать клавишу *Enter*. Имя объекта должно быть уникальным для данной модели и типа объекта.

Для того, чтобы изменить имя объекта, необходимо произвести любое из перечисленных действий:

- выполнить команду контекстного меню *Rename*;
- нажать клавишу *F2*;
- щелкнуть на объекте левой кнопкой мыши для его выделения, а затем щелкнуть на нем еще раз левой кнопкой мыши для редактирования имени.

Для того, чтобы создать сразу несколько объектов заданного типа, необходимо сделать следующее:

- выбрать нужный тип объекта на панели инструментов;
- переместить курсор мыши в нужную область модели;
- нажать клавишу *Ctrl* и щелкнуть кнопкой мыши на модели в местах расположения объектов;

- после последнего щелчка отпустить клавишу *Ctrl*.

При создании сразу нескольких новых объектов им сначала будут присвоены имена по умолчанию в соответствии с их типами.

При задании имени объекта, которое уже существует в базе данных для объекта такого же типа, система сообщит о неуникальности имени и предложит создать либо новый экземпляр существующего объекта, либо новый объект с таким же именем (рис. 51).

Можно:

- создать новый объект с таким же именем (опция *Create a new object with this name*);
- использовать существующий объект (опция *Use existing object*);
- выбрать вариант из списка и нажать кнопку *OK*
- или нажать кнопку *Cancel* и вернуться в окно моделирования для того, чтобы изменить имя объекта на другое.

Для того, чтобы *скопировать объект*, т.е. создать еще один экземпляр существующего объекта, нужно выделить его и выполнить команду *Copy* контекстного меню или выбрать пункт меню *Edit/Copy*. После этого необходимо выполнить команду *Paste (Вставить)* контекстного меню или команду меню *Edit/Paste*. В результате появится в модели образ скопированного объекта, и его можно будет переместить в любое другое место.

Важно отметить, что при помощи копирования нельзя создавать новые объекты, можно лишь повторно использовать уже существующие.

Для того, чтобы в модели переместить объект на другое место, нужно:

- выделить этот объект;
- нажать на нем левую кнопку мыши и, не отпуская ее, переместить объект на новое место;
- отпустить кнопку мыши.

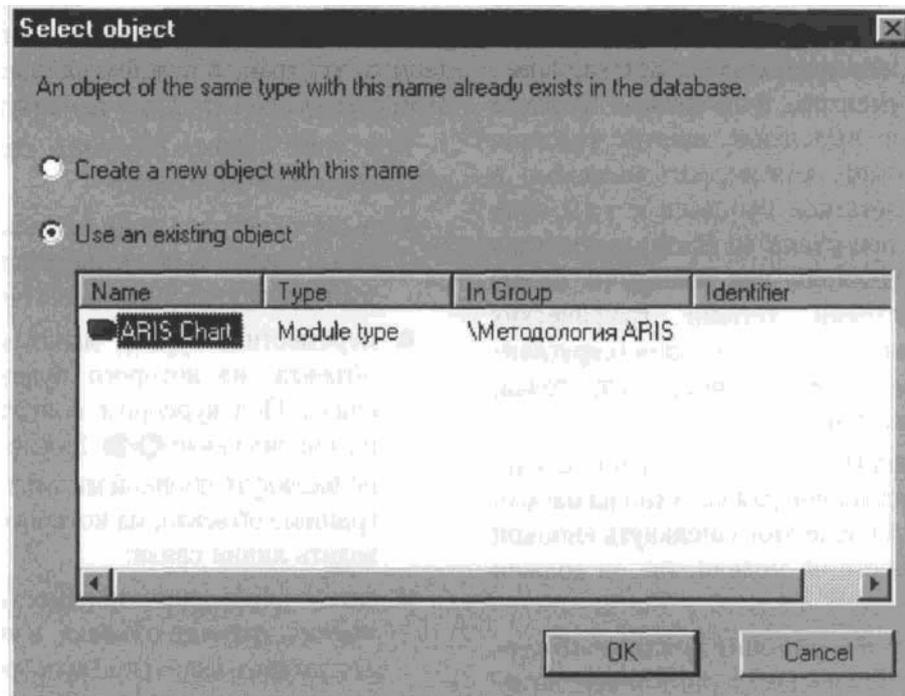


Рис. 51. Сообщение системы при создании объекта с неуникальным именем

При необходимости переместить объект в невидимую часть экрана следует медленно подвести его к границе экрана и затем двигать дальше. При этом скроллинг начнет постепенно прокручивать экран до нужного места. Можно также подвести объект к краю экрана, курсором мыши прокрутить скроллинг окна, а затем перемещать его дальше.

Если нужно переместить или скопировать группу объектов, то сначала следует выделить объекты этой группы любым из приведенных способов:

- поместить курсор мыши за пределами объектов. Нажать левую кнопку мыши и, не отпуская ее, вести курсор так, чтобы захватить область выделения всю группу объектов, а затем отпустить кнопку мыши;
- нажать клавишу *Ctrl* и, не отпуская ее, щелкать левой кнопкой мыши по очереди на каждом выделяемом объекте. После чего отпустить клавишу *Ctrl*,

Выделенные объекты обрамляются черными квадратами по углам и середине сторон. Группа перемещается тем же способом, что и одиночный объект. Можно также выделить группу объектов, нажать клавишу *Ctrl* и, нажав левую кнопку мыши, перемещать образ копируемой группы в нужное место модели. Затем отпустить клавишу и кнопку мыши, после чего группа объектов будет скопирована на указанное место.

Для вставки в модель *текстовых комментариев* нужно воспользоваться пунктом меню *Insert/Free-Form Text (Вставить текст в свободной форме)* или соответствующей кнопкой на панели инструментов. После выбора команды или нажатия кнопки необходимо щелкнуть левой кнопкой мыши на том месте модели, где должен быть расположен текстовый комментарий. В только что созданный объект вводится текст и нажимается клавиша *Enter*. Для того, чтобы в последствии изменить текст, в контекстном меню выбирается команда *Edit*.

### Моделирование бизнеса. Методология ARIS

Для включения в модель *графических объектов* можно воспользоваться кнопками панели инструментов. Графические объекты применяют для выделения цветом участков модели (например, критических моментов в процессе, недостатков процесса и т.д.). Они размещаются под (*Send to Back*) основными объектами модели или над (*Bring to Front*) ними. Доступными типами графических объектов являются прямоугольник (скругленный, нескругленный), эллипс, круг, точка, многоугольник, линия.

Для создания графического объекта следует выбрать подходящий для него тип на панели инструментов и после этого щелкнуть кнопкой мыши в том месте на модели, где он должен находиться.

Для того, чтобы изменить *размер объекта*, используются восемь черных точек, которые появляются вокруг объекта при его выделении. Нажав левую кнопку мыши на любой

из восьми точек и не отпуская кнопки, можно двигать эту точку в нужном направлении, при этом размер объекта будет изменяться.

Для того чтобы, *создать связь* между объектами модели, следует:

- выбрать на панели инструментов кнопку связи, если она не активизирована в данный момент (по умолчанию она активна);
- переместить курсор мыши на границу объекта, из которого будет выходить связь. Под курсором появится символ источника связи 0"в После этого нужно щелкнуть кнопкой мыши на том месте границы объекта, из которого будет выходить линия связи;
- затем необходимо переместить курсор мыши к границе объекта, в который будет входить связь (под курсором появится символ приемника связи **43**), и щелкнуть кнопкой мыши в нужном месте границы.

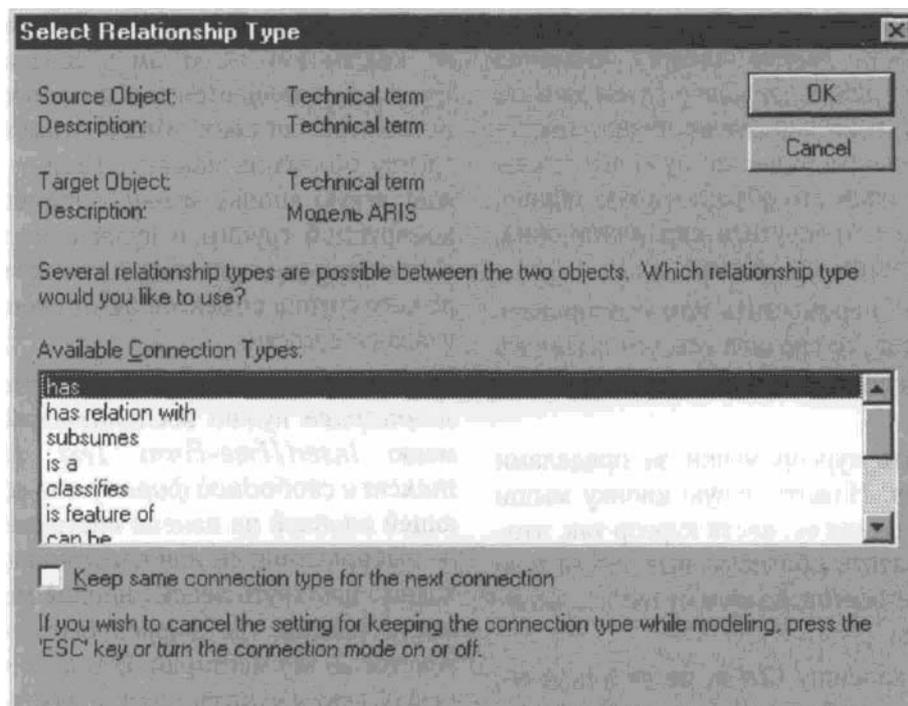


Рис. 52. Выбор типа новой связи

## Инструментальная система ARIS

После этих действий на экране появится окно с запросом типа новой связи (рис. 52). В этом окне в списке *Available Connection Types* (*Возможные типы связей*) находятся все связи, доступные при текущем фильтре. Выбрав нужный тип связи, следует нажать кнопку *OK*. Если этот тип необходим для всех новых связей, исходящих из текущего объекта к нескольким другим объектам, то надо включить опцию *Keep same connection type for the next connection* (*Сохранить этот тип связи для последующих соединений*). Для отключения этой опции нужно нажать клавишу *Esc*.

Управлять в модели расположением объектов можно либо командами меню *Arrange* (*Расставить*), либо кнопками панели инструментов:

- *Bring to Front* (*Поместить над*) — поместить выбранный объект или связь над всеми другими объектами;
- *Send to Back* (*Поместить под*) — поместить выбранный объект под всеми другими объектами модели;
- *Bring Forward* (*Перенести выше*) — поместить выбранный объект на один уровень выше;
- *Send Backward* (*Поместить ниже*) — поместить выбранный объект на один уровень ниже;
- *Align* (*Выровнять*) — выровнять выбранные объекты относительно левой (*Left*), правой (*Right*), верхней (*Top*) или нижней (*Bottom*) границы, либо относительно горизонтальной или вертикальной линий центра первого из группы выбранных объектов;
- *Equal Spacing* (*Равные расстояния*) — установить расстояния между объектами группы (по вертикали — *Vertical*, по горизонтали — *Horizontal*), равные расстояния между первыми двумя выбранными объектами группы;
- *Align to Grid* (*Выровнять по сетке*) — выровнять выделенные объекты по сетке;

- *Group* (*Группировать*) — объединить в группе выделенные объекты. После этого можно копировать и перемещать их как один объект;

- *Ungroup* (*Разгруппировать*) — разгруппировать выбранную группу или группы.

Если при создании модели произошла путаница в объектах и их связях, то можно воспользоваться пунктом меню *Edit/Layout* (*Расположение*). Система постарается помочь представить модель в более наглядном виде. Настройки параметров операции автоматического размещения объектов можно изменить с помощью меню *View/Option*, закладка *Layout*.

*Декомпозиция объекта* позволяет определять для объекта текущей модели ссылки на другие модели, детализирующие этот объект. Для того, чтобы декомпозировать объект на модель, следует:

- выбрать нужный объект и вызвать на экран окно с его свойствами (пункт контекстного меню *Properties*);

- выбрать закладку *Assignments* (*Декомпозиция*). Эта форма содержит уже определенные для объекта модели декомпозиции. Просмотреть их можно при помощи команды контекстного меню *Open*. Для изменения свойств модели выбирается пункт контекстного меню *Properties*;

- для определения новой модели необходимо нажать кнопку *New*. На экране появится окно *Assignment Wizard* (*Помощник декомпозиции*) для определения новой, декомпозирующей этот объект модели. В нем выбирается кнопка *Далее*, после этого открывается следующее окно;

- в окне *Assignment Wizard* можно определить новую модель (*New Model*) или воспользоваться уже существующими (*Existing Model*). Для создания новой модели нужно выбрать в списке тип модели и нажать кнопку *Далее*, после чего откроется второе окно *Assignment Wizard*, в котором нужно указать местонахождение модели.

После выбора группы и нажатия кнопки *Готово* на экране откроется окно для создания новой модели. При использовании существующей модели нужно выбрать в списке требуемый тип модели и нажать кнопку *Далее*. На экране появится окно выбора модели. После выбора нажимается кнопка *Готово*.

*Распечатать модель* можно командой меню *File/Print* или соответствующей кнопкой на панели инструментов. Для предварительного просмотра печати пользуются командой меню *File/Print Preview* или кнопкой на панели инструментов. Настройка формата печати находится в окне настроек — меню *View/ Options*, закладка *Print Options (Опции печати)*.

## 5.4. Атрибуты элементов ARIS — ARIS Attributes

Модуль *ARIS Attributes* (также известный как *Окно Attributes*) дает возможность легко изменять и записывать характеристики баз данных и всех их элементов (рис. 53).

Каждый элемент базы имеет набор атрибутов. С их помощью можно задать количественные и качественные характеристики моделируемой деятельности. В частности, для каждого объекта можно задать имя, уникальный код в проекте, автора, время и дату создания, детальное описание, пример реализации, затраты времени и средств на выполняемую функцию и т. д. Все указанные характеристики объектов и связей формализованы и используются при проведении анализа или составлении отчета.

Атрибуты элементов ARIS собраны в иерархические папки (аналогичные папкам в *ARIS Explorer*). В каждой папке находятся атрибуты одного типа. Например, в папке *Change Management (Управление изменениями)* собраны атрибуты, которые используются в модуле *ARIS Change Management* для управления изменениями моделей.

Каждый элемент имеет свой набор атрибутов, однако среди них существует довольно много общих.

Значения атрибутов заносятся в соответствующие таблицы. Данные можно перемещать между ячейками таблицы или копировать, используя механизм *Drag and Drop (Перетащить и оставить)*.

Автоматически задаваемые системой данные, которые не могут изменяться вручную, помещены в ячейки с серым фоном для того чтобы легко отличить их от изменяемых данных. Например, два атрибута со значением *Last change (Последнее изменение)* и *Last user (Последний пользователь)* будут автоматически модифицированы системой при каждом выполненном изменении, которое сохраняется в процессе редактирования атрибута.

Когда открывается окно свойств элемента, то закладка *Maintained Attributes (Заполненные атрибуты)* покажет только те атрибуты, которые поддерживаются системой (рис. 54). Отображенные данные можно полностью или частично копировать, вставлять в другие приложения и использовать для сообщений.

К функциям модуля *ARIS Attributes* относится наследование атрибутов (*Inherit Attributes*). Эта функция запускается посредством одноименного пункта контекстного меню, вызванного на выделенном элементе.

При наследовании в системе ARIS значения выбранных атрибутов одного элемента присваиваются атрибутам другого элемента. Могут быть наследованы атрибуты баз данных, папок, объектов, моделей, соединений, пользователей и их групп, языков и форматов шрифтов. Не может быть унаследован атрибут *Name (Имя)*. Для проведения наследования необходим привилегированный доступ к базам данных (см. раздел 9.8).

Механизм наследования реализуется при помощи окон *Attributes Inheritance Wizard (Помощник наследования атрибутов)*, показанных на рис. 55 - 56. В первом окне (рис. 55) предлагается выбрать атрибуты, которые необходимо наследовать.

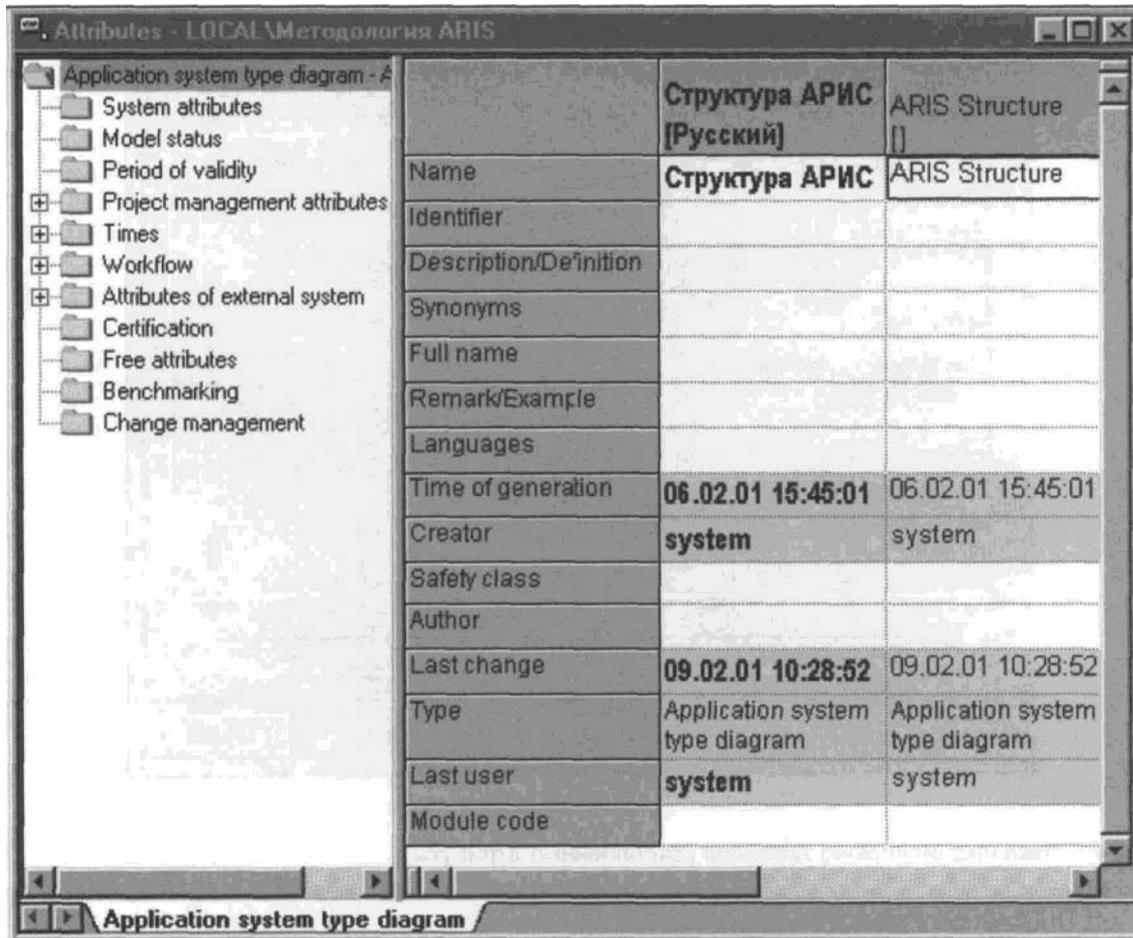


Рис. 53. Окно Attributes. Просмотр и редактирование атрибутов

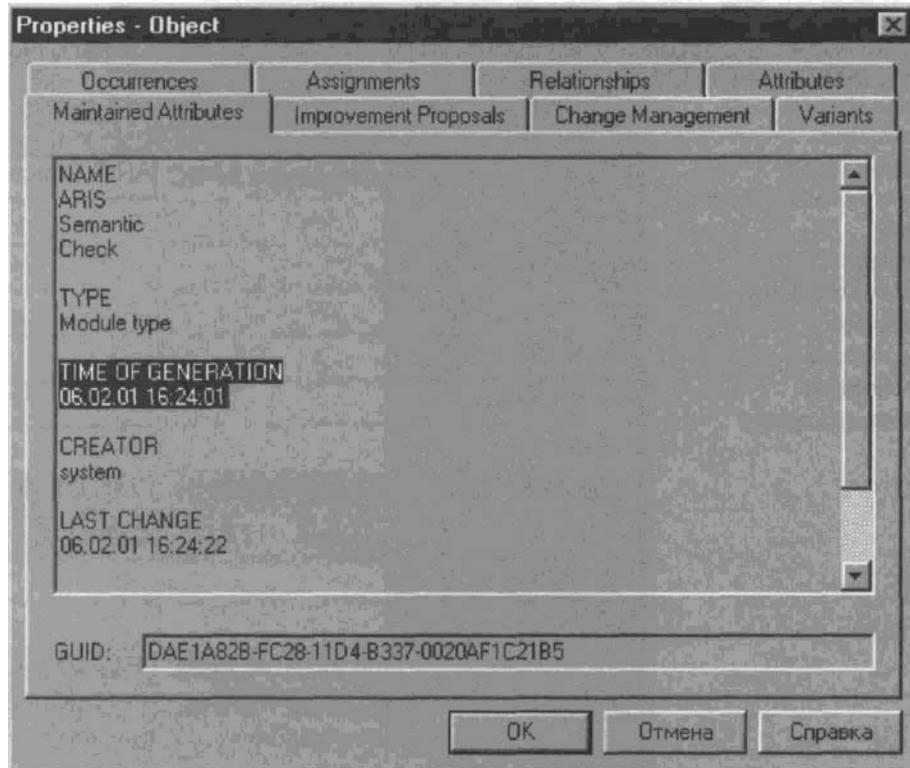


Рис. 54. Закладка Maintained Attributes (Заполненные атрибуты)

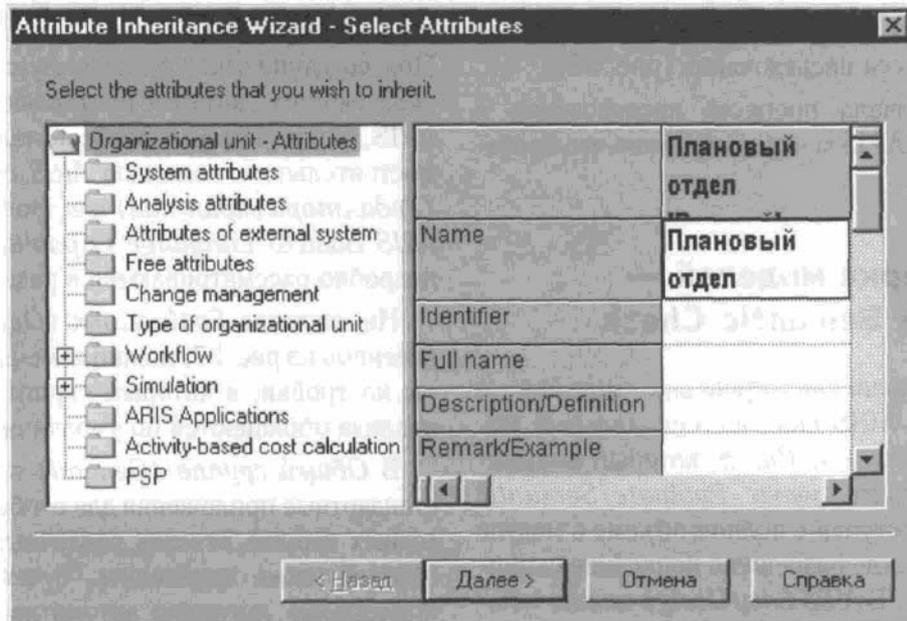


Рис. 55. Окно Attribute Inheritance Wizard. Выбор наследуемых атрибутов

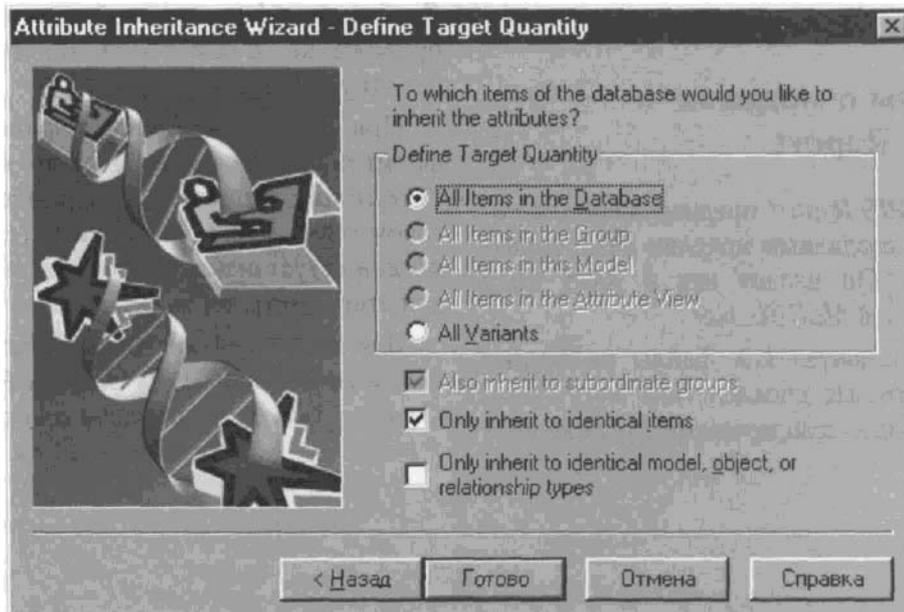


Рис. 56. Окно Attribute Inheritance Wizard. Выбор опций процесса наследования

Во втором и последнем окне выбираются опции процесса наследования (рис. 56).

После начала процесса наследования в окне вывода ARIS можно наблюдать его протекание.

## 5.5. Поверка моделей — ARIS Semantic Check

Для выполнения проверок моделей и объектов в ARIS включен специальный модуль *ARIS Semantic Check*, который вызывается из пункта меню *Evaluate/Semantic Check*. Он доступен в полном объеме в модуле ARIS Toolset, где разрешено написание скриптов проверок. В ARIS Easy Design можно только осуществлять проверки моделей.

Работа с этим модулем, применяемые правила проверок моделей, создание собственных правил проверок подробно рассмотрены в разделе 9.6.

## 5.6. Отчеты о моделях — ARIS Report

Модуль *ARIS Report* предназначен для составления по созданным моделям разнообразных отчетов. Он входит как в *ARIS Easy Designer*, так и в *ARIS Toolset*.

Отчеты создаются как файлы различных форматов, которые впоследствии могут быть отредактированы стандартными приложения

ми типа *Word*, *Excel* или редактором *HTML*. При создании отчетов использует так называемые скрипты, которые поставляются вместе с ARIS, или те, которые пользователь создал самостоятельно с помощью *ARIS Script Editor (Редактора скриптов)* и встроенного языка *ARIS Basic® Language Capability*. Скрипты подробно рассматриваются в разделе 10.

На закладке *Evaluations (Оценка)*, приведенной на рис. 57, можно изменять параметры настройки, к которым скрипты отчетов и анализа обращаются по умолчанию.

В *Общей группе (General)* определяются стандартные приложения для отображения выходных файлов отчетов различных форматов. Можно также определить, будет ли каждое приложение запущено автоматически и будет ли графика интегрирована в формат *HTML* как файлы формата *GIF* или *JPG*.

В списке отображены все форматы файла, которые могут использоваться для отчетов. Для каждого формата необходимо определить приложение, с помощью которого файлы будут открыты.

При назначении нового приложения для формата файла нужно нажать кнопку *Edit (Редактирование)*. В открывшемся диалоговом окне *Select Application (Выбор приложения)* становятся все необходимые для этого настройки.

Для получения сообщения после генерации файла отчета ставиться флажок *Prompt after Creation (Сообщение после создания)*.

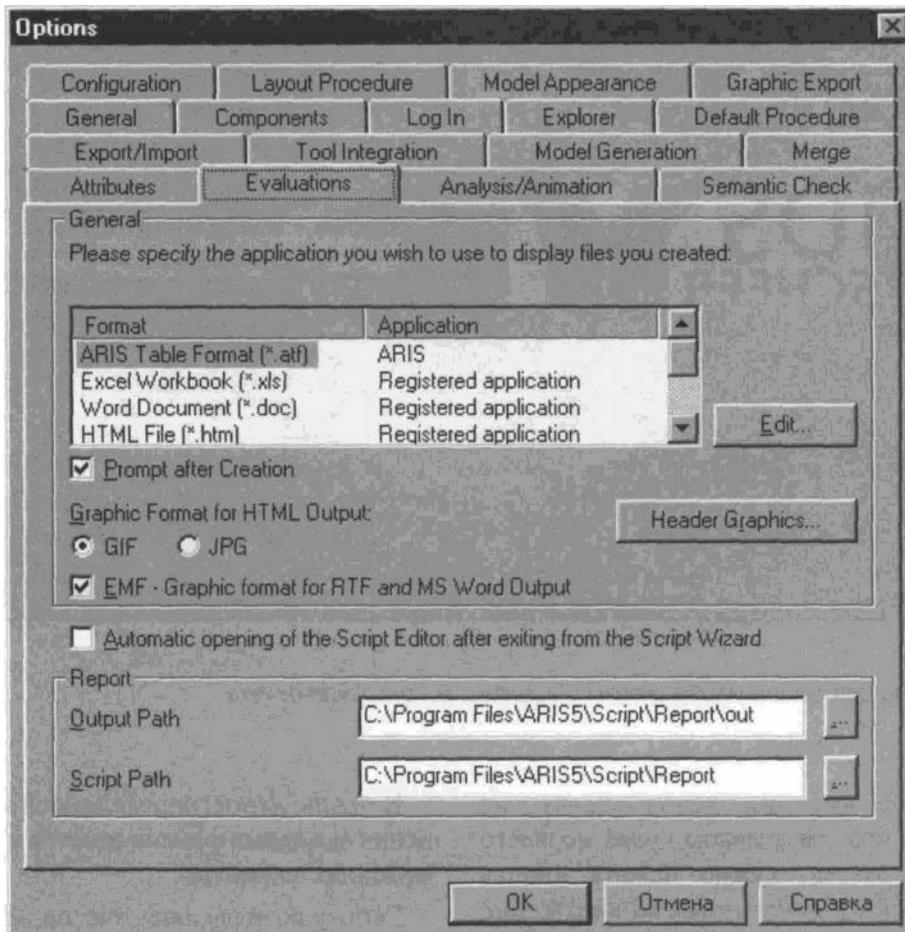


Рис. 57. Закладка Evaluations окна Options

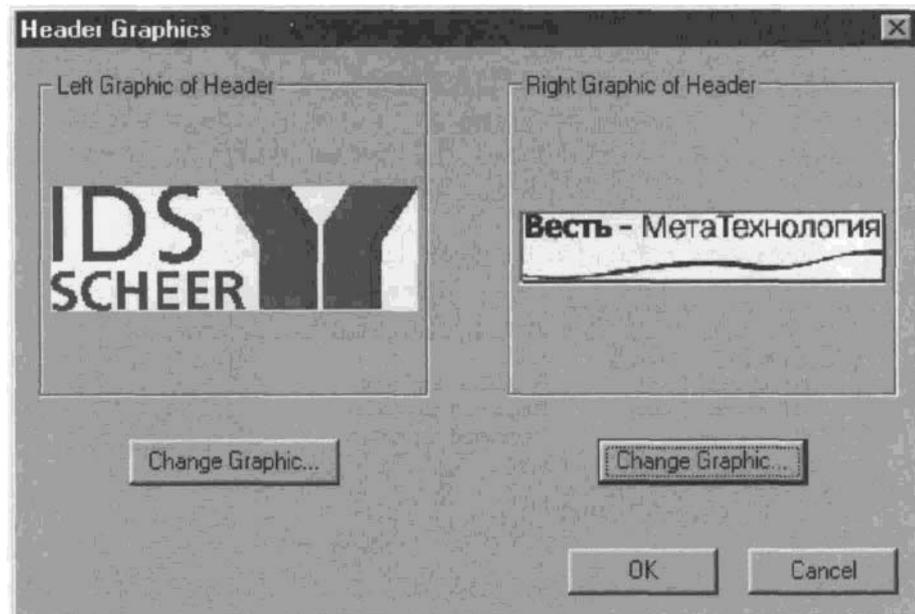


Рис. 58. Окно для смены изображений в верхнем колонтитуле отчета

Для замены эмблем, используемых по умолчанию в правом и левом краях верхнего колонтитула отчета, нужно нажать кнопку *Header Graphics* (*Графическое изображение в колонтитуле*). Принятое графическое изображение будет отображено (рис. 58) в диалоговом окне *Header Graphics*. С помощью кнопок *Change Graphic* (*Изменить графическое изображение*) можно выбирать другие эмблемы.

Если в отчетах формата RTF или DOC требуется использовать графику формата EMF, то нужно поставить флажок EMF — Graphics Format for RTF and MS Word Output (Использование графического формата EMF для вывода отчетов в форматах RTF и DOC). Если флажок не выбран, то применяется графика формата WMF.

Поставленный флажок *Automatic opening of the Script Editor after exiting from the Script Wizard* приводит к автоматическому открытию приложения, редактирующего созданный отчет при выходе из окна *Script Wizard* (*Помощник создания отчетов*).

В группе *Report* определяются места размещения выходных файлов отчетов и файлов, содержащих скрипты.

Работа по созданию отчетов подробно рассматривается в главе 10.

## 5.7. Анализ и анимация моделей — APIS Analysis

*ARIS Analysis* так же как *ARIS Simulation* и *ARIS ABC*, является модулем оценки моделей. *ARIS Analysis* вызывается через пункт меню *Evaluate/Analysis* (*Оценка/анализ*) или через аналогичный пункт контекстного меню.

Разработанные модели могут быть проанализированы по определенным правилам, которые задаются в специально разработанных алгоритмах — скриптах, написанных на языке, схожем с языком *Visual Basic*. В среде ARIS существуют стандартные алгоритмы анализа:

### Инструментальная система ARIS

классификация функций в зависимости от значений их атрибутов. Данный алгоритм построен так, что он может быть настроен для различных нужд (например, для выявления количества выполняемых вручную функций или функций, добавляющих стоимость к конечному продукту);

- *затраты времени и средств на выполнение процесса.* Существует возможность в рамках модели процесса задать стоимость и продолжительность выполнения отдельных его шагов, а также указать продолжительность переходов от одного шага к другому. Эта информация учитывается при проведении анализа данного вида;

- *необходимые организационные изменения в процессах.* Данный вид анализа позволяет получить рекомендации по изменению ответственности исполнителей за выполнение отдельных операций процесса;

- *анализ движения информации внутри процесса.* Выявляются информационные разрывы внутри смоделированных процессов;

- *анализ использования элементов автоматизированной системы в процессах.* Определяется состав функ

- классификация функций — анализ и ций, поддерживаемых автоматизированной системой и выполняемых традиционным способом;

Результаты анализа представляются в виде таблиц, которые могут быть просмотрены как встроенными средствами ARIS, так и другими приложениями (редакторами *RTF*, *Microsoft Word*, *Excel* и т. п.).

Пользователи имеют возможность создавать собственные алгоритмы анализа, которые будут применяться в последующих проектах. Структура и состав отчета определяются используемым алгоритмом, который также может быть разработан пользователем самостоятельно.

Более подробно стандартные алгоритмы анализа ARIS рассмотрены в разделе 9.7.

Модуль *ARIS Analysis* дает возможность реализовать некоторые элементы анимации моделируемых процессов. Для этого в модуль введен инструмент, с помощью которого пользователь может отображать динамическую логику процесса и на основе соответствующих атрибутов объектов, вычислять индексы, пропорциональные времени или ориентировочной стоимости процессов.

Для управления анимацией служит панель инструментов *Animation Control* (*Управление анимацией*), приведенная на рис. 59.



Рис. 59. Панель инструментов Animation Control

Кнопки панели инструментов позволяют:

-  создать новый скрипт анимации модели;
-  открыть существующий скрипт анимации модели;
-  сохранить текущий скрипт анимации;
-  выйти из режима анимации;
-  переключиться в режим проигрывания анимации;
-  переключиться в режим записи скрипта анимации;
-  запустить скрипт анимации заново;
-  остановить процесс анимации;
-  осуществлять пошаговую анимацию;
-  включить полуавтоматический режим анимации;
-  включить автоматический режим анимации;
-  воспользоваться связью по значению «истина»;
-  воспользоваться связью по значению «ложь».

Анимация дает возможность произвести анализ выделенного в модели пути процесса с точки зрения временных и стоимостных характеристик. Она может быть выполнена для всех типов моделей, включающих функции, события, описания правил и процессов взаимодействия объектов.

В течение анимации процесс последовательно проходит несколько заранее заданных шагов — так называемый скрипт анимации. Этим можно управлять вручную, автоматически или полуавтоматически. Сначала нужно написать скрипт анимации, т.е. описать алгоритм прохода маркера по объектам модели. Для этого нужно открыть модель, для которой будет написан скрипт прохода маркера.

Создание нового скрипта инициируется выбором пункта меню *Animation/New*. После этого на экране появится окно статистики этого процесса, и система предложит начать запись скрипта, выделив первый объект модели.

Если использование выделенного объекта в анимации подтверждается, то выбирается пункт меню *Animation/Yes*, в ином же случае

— *Animation/No* (можно нажать кнопки со знаками «V» и «X» панели инструментов). Затем нужно выбрать следующий объект, также выделив его курсором, и т.д.

После того, как пройдена вся цепочка анимации, выбирается пункт меню *Animation/End* (*Анимация/Окончание*) или *Animation/Save* (*Анимация/Сохранить*), и скрипт сохраняется.

Для того, чтобы «проиграть» существующий скрипт прохода маркера по модели, нужно:

- открыть модель, для которой был написан скрипт;
- выбрать пункт меню *Animation/Open* (*Анимация/Открыть*);
- выбрать из списка нужный скрипт и нажать кнопку *OK*.

После этого скрипт будет загружен, и можно начать его выполнение, используя команды меню или кнопки панели инструментов:

- *Automatic* — автоматическое выполнение скрипта;
- *Step* — пошаговое выполнение;
- *Pause* — пауза в выполнении скрипта;
- *Reset* — прервать выполнение и начать заново.

При этом в окне процесса будут отображаться текущие значения стоимостных и временных атрибутов объектов модели. Для выхода из режима анимации выбирается пункт меню *Animation/End* или соответствующая кнопка на панели инструментов.

## 5.8. Управление изменениями — APIS Change Management

Моделирование бизнеса только тогда является эффективным, когда построенные модели адекватно описывают бизнес-процессы организации. Адекватность достигается не сразу, и поэтому модели процессов нуждаются в постоянном совершенствовании. Для этой цели в ARIS включен модуль *ARIS Change Management*. Он реализует механизм сбора, хранения, учета и обобщения предложений по улучшению объектов и модели

## Инструментальная система ARIS

Предложения об изменении и совершенствовании могут быть сделаны для всех объектов и моделей. Они регистрируются на специальных полях закладки *Change Management* в диалоговом окне *Properties* (рис. 60). Пользователи моделей также могут вводить изменения и предложения по совершенствованию путем редактирования атрибута *Change Management* объектов и моделей базы данных.

В то же время менеджер процесса посредством элемента базы данных *Improvements (Изменения)*, может получать информацию обо всех предложениях по совершенствованию различных моделей и объектов. В закладках *Change Management* менеджер может назначать мероприятия для разработчиков. Разработчики модели получают список предназначенных для них задач через папку базы данных *Tasks (Задачи)*.

Модуль *ARIS Change Management* является важным инструментом для документирования и анализа процессов с точки зрения концепции непрерывного совершенствования (*Continuous Process Improvement*). Он незаменим при построении и эксплуатации систем менеджмента качества в соответствии с международными стандартами ИСО серии 9000 (см. раздел 11.2).

Механизм улучшения может быть запущен, если все пользователи моделей будут вводить свои предложения по совершенствованию непосредственно в базу данных ARIS, консультируясь или нет с менеджером процесса. Предложения собираются менеджером, после чего модели модифицируются разработчиками в соответствии с его указаниями. Менеджер процесса ответствен за фактическое выполнение изменяемых процессов. Таким образом, смоделированные процессы будут

оценены, и мероприятия по их изменению будут проводиться непосредственно в том месте, где они фактически выполняются.

Порядок работы с модулем *ARIS Change Management* состоит в следующем. Менеджер конкретной модели, например, «Знания о менеджменте качества» (см. раздел 7), выделяет объект «Официальные документы ИСО», после чего открывает окно с его свойствами, а в нем — закладку *Change Management* (рис. 60).

На ней нужно:

- выбрать язык (список *Language*);
- ввести предложения (поле *Proposal*);
- наметить мероприятия по их выполнению (*Measures*);
  - при необходимости сделать замечания (*Remark*);
  - установить приоритет мероприятий (*Priority*);
  - назначить дату выполнения мероприятий (поле *End Date*) при помощи раскрывающегося календаря (кнопка *t*);
  - назначить один из четырех возможных статусов мероприятий (список *Status*):
    - Being processed (в процессе выполнения),
    - Done (сделано).
    - Not assigned (не установлен),
    - Postponed (отсроченный);
  - назначить ответственного из числа установленных пользователей и их групп (список *Responsibility (Ответственность)*);
  - при необходимости установить флажок сброса всех атрибутов *Change Management* по окончании внесения изменений (*Finished: Reset All Change Management Attributes*).

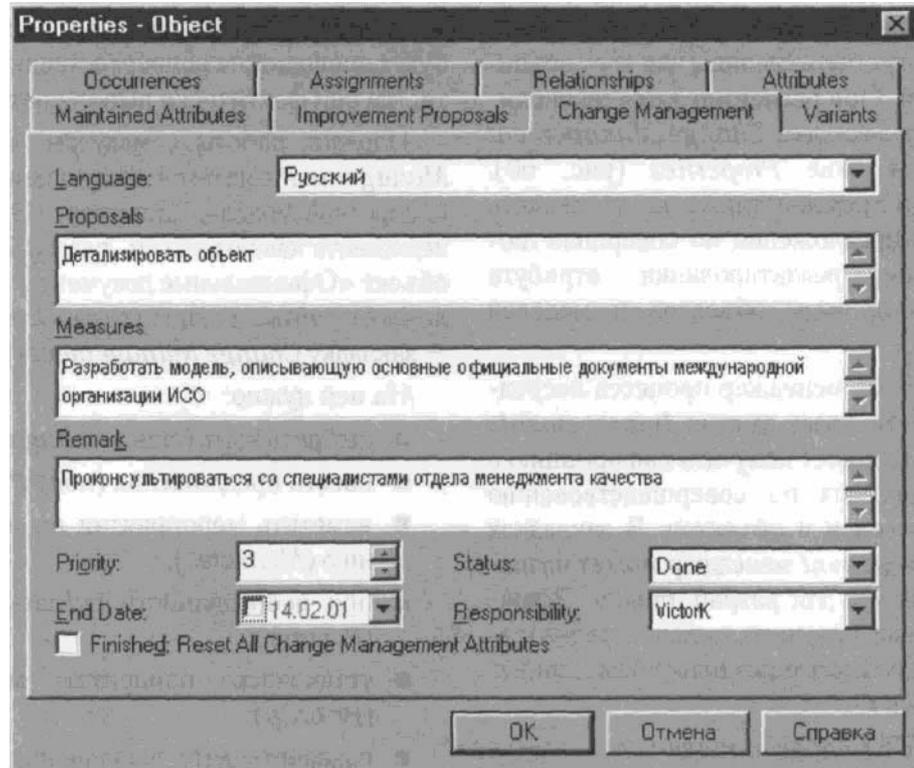


Рис. 60. Закладка Change Management (Управление изменениями)

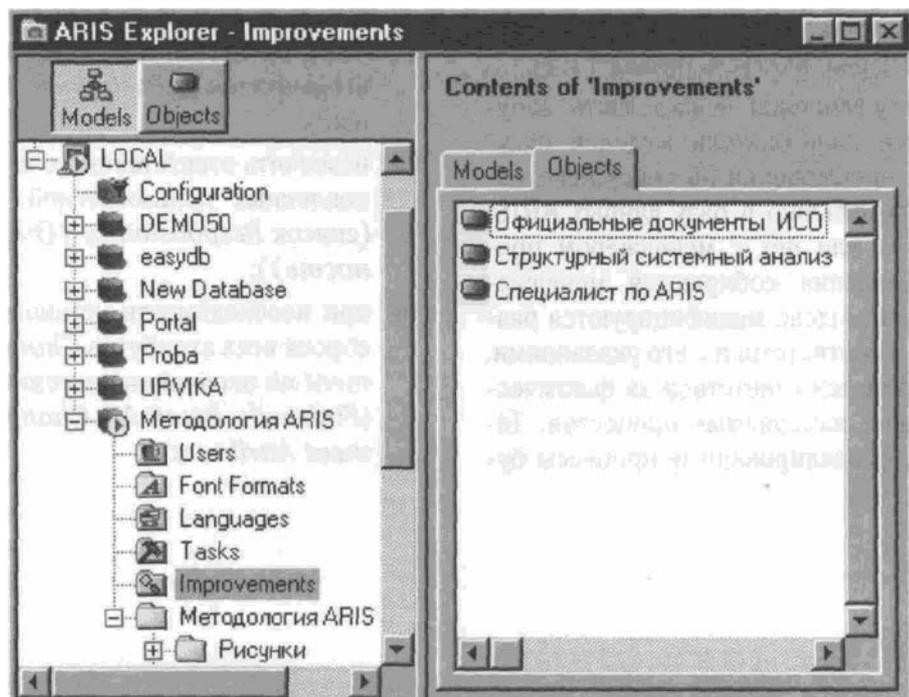


Рис. 61. ARIS Explorer. Содержание папки Improvements (Изменения)

Объекты и модели, с которыми проделаны перечисленные действия, появятся на закладках *Objects* и *Models* соответственно папок *Task (Задачи)* и *Improvements (Изменения)* используемой базы данных (рис. 61).

Информация, внесенная в поля закладки *Change Management* окна, содержащего свойства объекта, автоматически заносится в атрибуты *Change Management* объекта (рис. 61). Они могут быть просмотрены менеджером процесса традиционным образом непосредственно в папках *Task* и *Improvements*.

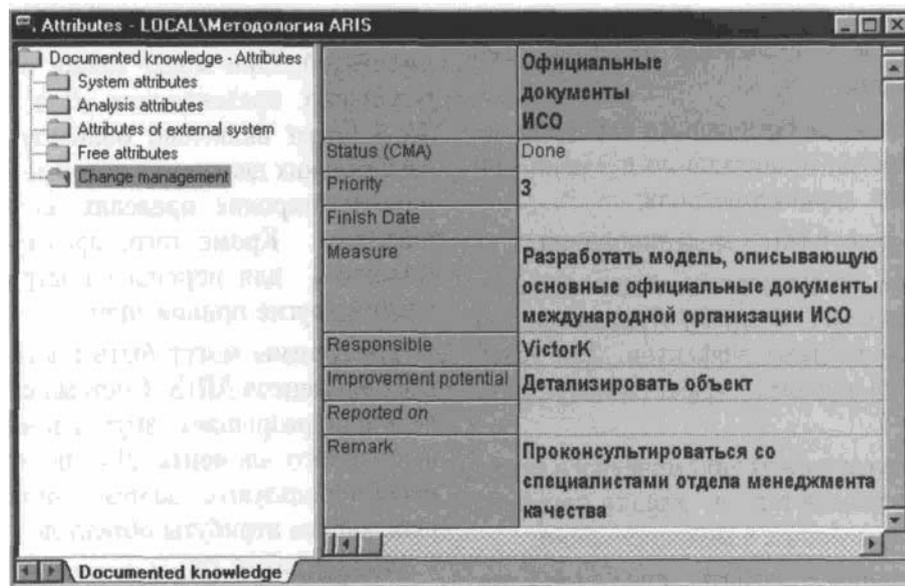


Рис. 62. Папка *Change Management* окна *Attributes* объекта «Официальные документы ИСО» модели «Знания о менеджменте качества» (см. раздел 7)

## 5.9. Имитационное моделирование — APIS Simulation

Модуль имитационного моделирования *ARIS Simulation* используется в тех случаях, когда необходимо моделировать поведение во времени разработанных моделей бизнес-процессов. Цель динамического моделирования — определение узких мест в реализации процессов (несогласованность параллельно выполняемых процессов, нехватка ресурсов для эффективного выполнения процессов и т. д.).

Разрабатывая модели бизнес-процессов, можно задать различные характеристики в качестве основы для моделирования во времени. Для функций определяют время ожидания, и

подготовки к работе, время и периодичность выполнения, а также элементарные показатели стоимости. Показатели стоимости выполнения функции могут быть взяты из результатов функционально-стоимостного анализа, проводимого в модуле *ARIS ABC*. Для событий, влияющих на выполнение функций, можно задать периодичность их появления. Перечисленные данные определяют реальные условия выполнения процесса во времени и могут успешно применяться для достаточно точного динамического моделирования, которое позволяет выявить преимущества и недостатки предлагаемых изменений. Анализируя альтернативные варианты реализации процессов и определяя их количественные характеристики (общее время на выполнение про-

цесса, коэффициенты использования трудовых ресурсов), можно отобразить вариант, наиболее эффективный с точки зрения выделенных критериев.

Процесс проведения анализа на основе динамического моделирования можно разбить на следующие этапы:

- создание моделей процессов;
- подготовка и занесение данных — временных и стоимостных характеристик процессов;
- проведение динамического эксперимента — прогон процессов во времени при заданных характеристиках;
- анализ результатов моделирования.

В процессе динамического моделирования реальное поведение объектов отражается при помощи анимационных эффектов. Для ускорения процесса подобные эффекты могут быть отключены.

Основные результаты динамического моделирования отражаются в характеристиках объектов, участвующих в моделировании:

- для функций — затраты денег и времени на реализацию, время ожидания освобождения ресурсов и т.п.;
- для точек разветвления — время ожидания завершения параллельных процессов;
- для организационных единиц — коэффициенты использования.

Собранная статистика о трудовых ресурсах, несогласованности параллельных процессов, наличии узких мест может быть как проанализирована средствами ARIS (в табличной и графической формах), так и экспортирована в другие средства (*Microsoft Excel*) для более детального анализа, построения дополнительных диаграмм и графиков.

## 5.10. Построение диаграмм и графиков — ARIS Chart

Модуль *ARIS Chart* предназначен для создания графических диаграмм, описывающих различные аспекты разработанных моделей. Такие диаграммы благодаря трехмерному представлению и использованию тени для объектов обеспечивают эффективный анализ

информации, заложенной в моделях, и их взаимосвязей. Они могут также применяться в различных презентациях. Для этой цели в *ARIS Chart* включено большое число типов графических диаграмм, которые можно изменять в широких пределах, создавая новые шаблоны. Кроме того, предусмотрена возможность для переноса построенных диаграмм в другие приложения.

Диаграммы могут быть построены не для всех элементов ARIS. Система сама разрешает или не разрешает запускать *ARIS Chart* для выбранного элемента. Для построения диаграмм используются данные, записанные в определенные атрибуты объектов.

Модуль *ARIS Chart* запускается при помощи контекстного меню, вызываемого на выделенных объектах модели. При этом появляется *Chart Wizard (Мастер построения диаграмм)* и его первое окно (рис. 63). В нем необходимо выбрать тип объекта, на котором будет основываться построение диаграммы.

После нажатия кнопки *Далее* появляется второе окно (рис. 64), где в выпадающем меню необходимо выбрать тип диаграммы *Chart Type*. Ее вид отражается в окне предварительного просмотра *Preview*.

В последнем окне *Chart Wizard* нужно выбрать описание диаграммы (*Chart Definition*). Стандартные описания диаграмм находятся в папке *Configuration (Конфигурация)* каждого сервера (рис. 65). Они могут быть изменены при помощи механизма построения определений диаграмм (*Chart Wizard*), который запускается посредством контекстного меню, вызываемого на закладке *Chart Definition (Описание диаграмм)*.

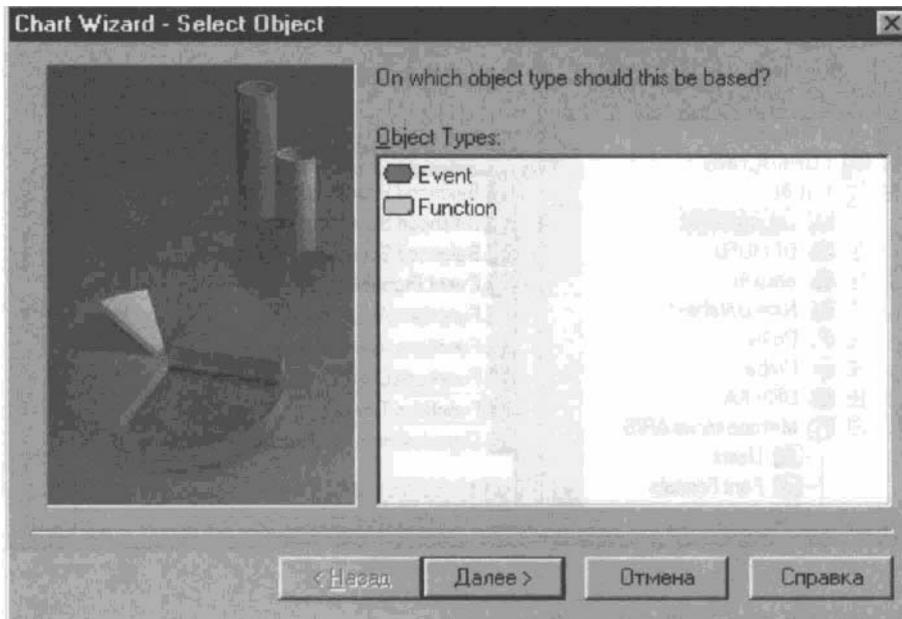


Рис. 63. Окно Chart Wizard. Выбор типа объекта для построения диаграммы

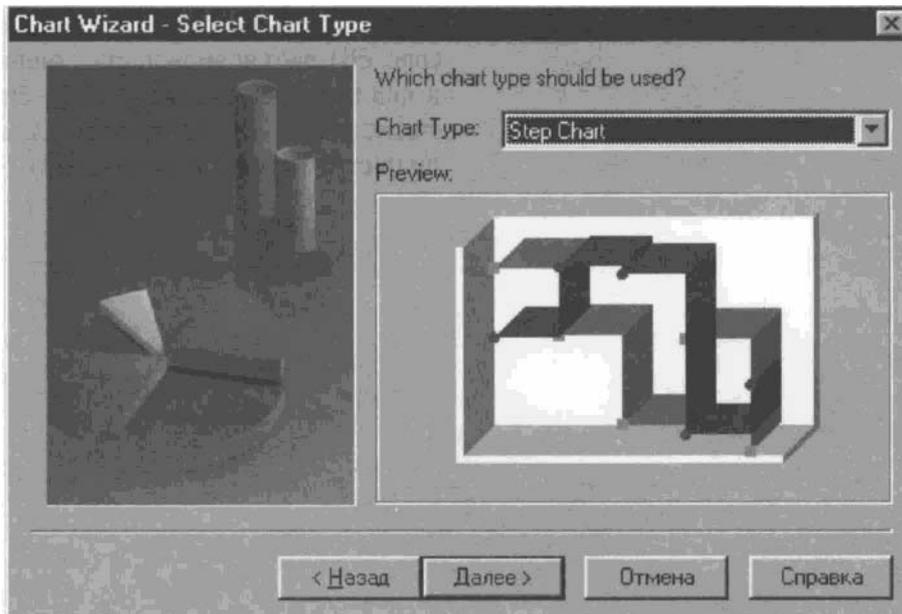


Рис. 64. Окно Chart Wizard. Выбор типа диаграммы

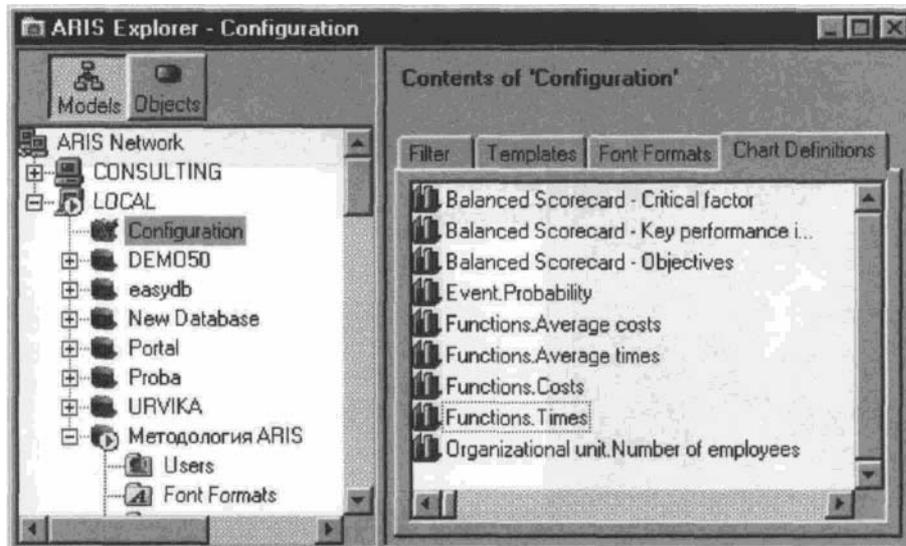


Рис. 65. Стандартные описания диаграмм

После выбора описания диаграммы нажимается кнопка *Готово* и появляется экран *ARIS Chart* (рис. 66) с построенной диаграммой (рис. 67).

На экране, кроме стандартных пунктов меню, появляется дополнительный пункт *Chart (диаграмма)*, содержимое которого (рис. 66), дает возможность изменять свойства и вид построенной диаграммы. Эти операции также могут быть проведены с помощью панели инструментов *Chart* (рис. 68).

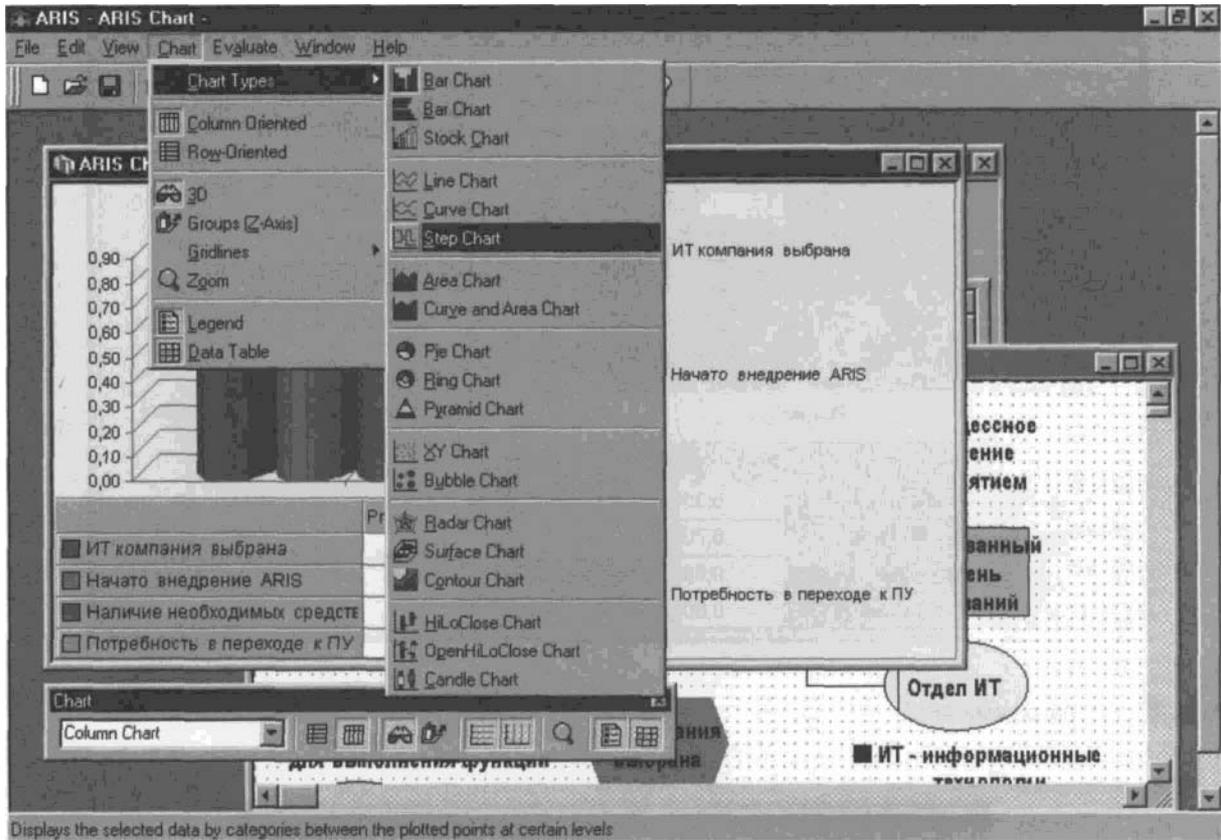


Рис. 66. Экран ARIS Chart

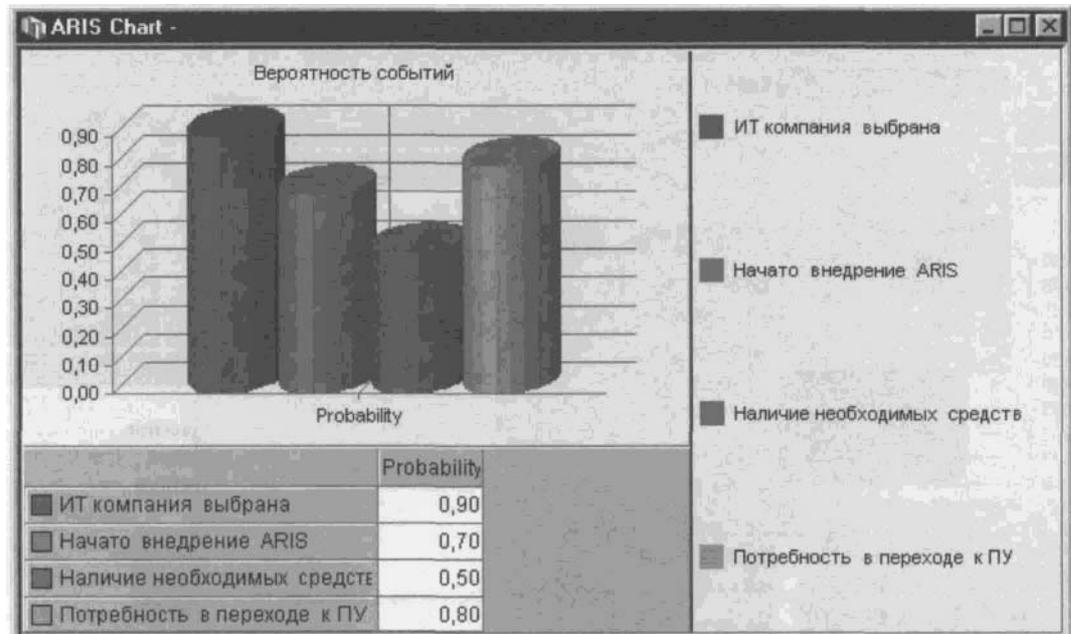


Рис. 67. Диаграмма ARIS



Рис. 68. Панель инструментов Chart

Назначение кнопок панели:

- выпадающий список типов диаграмм;
- горизонтальная ориентация диаграммы;
- вертикальная ориентация диаграммы;
- трехмерное изображение диаграммы;
- включение/выключение оси Z;

- включение/выключение горизонтальной сетки;
- включение/выключение вертикальной сетки;
- увеличение;
- включение/выключение описания диаграммы (легенда);
- включение/выключение таблицы.



Рис. 69. Закладка General (Общие свойства) ARIS Chart

Свойства *ARIS Chart* настраиваются с помощью окна *Свойства: ARIS Chart*, вызываемого через пункт меню *Edit/Properties*. Четыре закладки этого окна показаны на рис. 69 - 72. На закладке *General* (Общие свойства) в поле *Effects* (Эффекты) задается вид диаграммы, а в поле *Color* (Цвета) — тип палитры, цвета ее элементов и узоры.

На закладке *Data Series* (Серии данных) задается оформление отдельных серий данных — цвета, тип и толщина линий, тип диаграммы и ее параметры.

На закладке *Axis* (Оси) задается оформление осей диаграммы.

На закладке *3D* (Трехмерные эффекты) определяются параметры трехмерных эффектов диаграммы.



Рис. 70. Закладка Data Series (Серии данных)

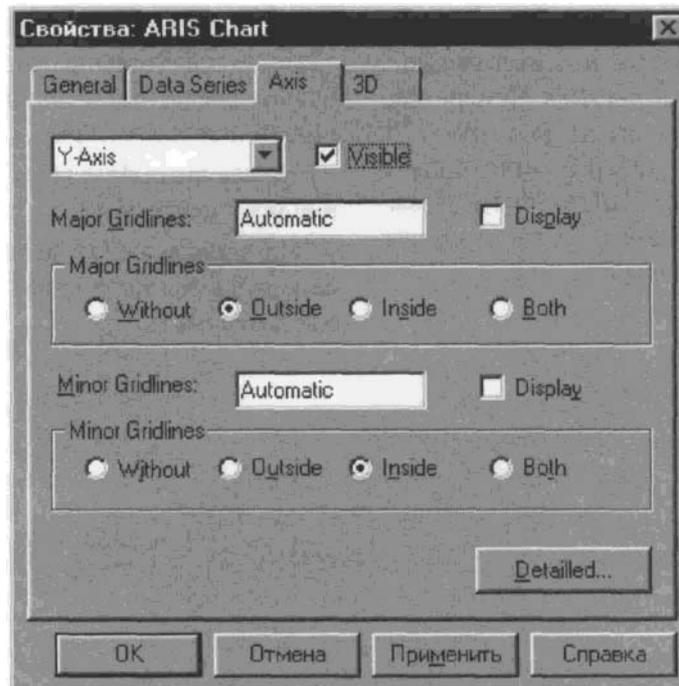


Рис. 71. Закладка Axis (Оси)

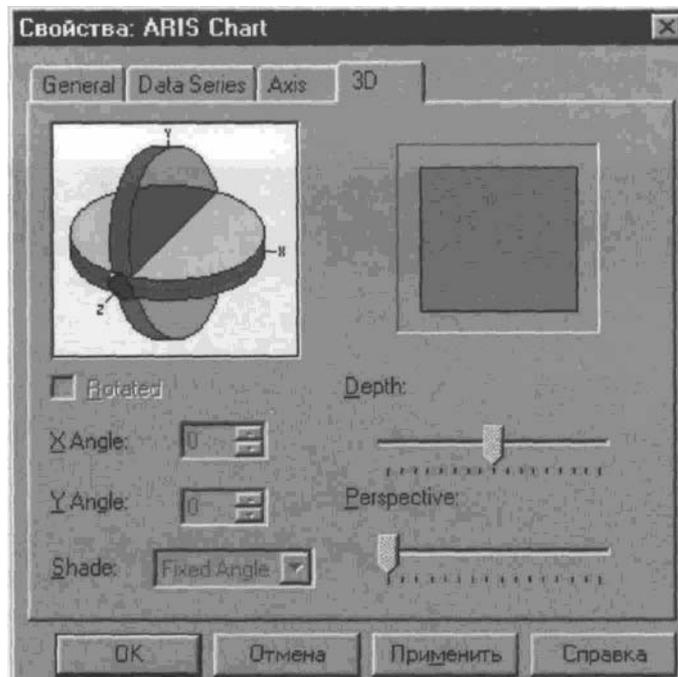


Рис. 72. Закладка 3D Трехмерные эффекты

## APIS Administrator

### 5.11. Администратор APIS —

Модуль *ARIS Administrator* предназначен для управления базами данных сервера и обмена ими с другими серверами. Он доступен и в модуле *ARIS Easy Design*, и в модуле *ARIS Toolset*.

После того, как модуль *ARIS Administrator* установлен, его свойства можно найти в свойствах серверов и баз данных на закладке *Administration (Администрирование)* (рис. 73 - 74).

С помощью *ARIS Administrator*, привилегированный пользователь может выполнять следующие действия с серверами (рис. 73):

- изменять пароли администраторов сервера и конфигурации (кнопка *Change Password (Изменить пароль)*);
- показать подключенные к данному серверу базы данных (кнопка *Connections (Соединения)*);

• делать доступными базы данных, импортируя их с другого сервера и регистрируя на текущем сервере. Для этого используются кнопка *Restate (Восстановить)* (файлы формата *ARIS Compressed Database*, или *\*.acg*) и кнопка *Register (Регистрировать)* (файлы формата *Poet index/lie*, или *\*.idx*). База данных в формате *^acd* является резервной копией, и может использоваться для переноса на другой сервер. Функции *ARIS Administrator* позволяют управлять различными базами данных, расположенными на сервере ARIS. Имеются возможности (рис. 74):

- отменять регистрацию базы данных на сервере (кнопка *Deregister*);
- создавать резервные копии баз данных в виде файлов формата *\*.acd* (кнопка *Backup (Создание резервных копий)*);
- очищать базы данных от неиспользуемых и двойных элементов (кнопка *Reorganize (Преобразовать)*).

**Моделирование бизнеса. Методология ARIS**

Модуль *ARIS Administrator* в отличие от модуля *ARIS Merge* дает возможность переносить информацию о пользователях и их группах от исходной базы данных к целевой базе

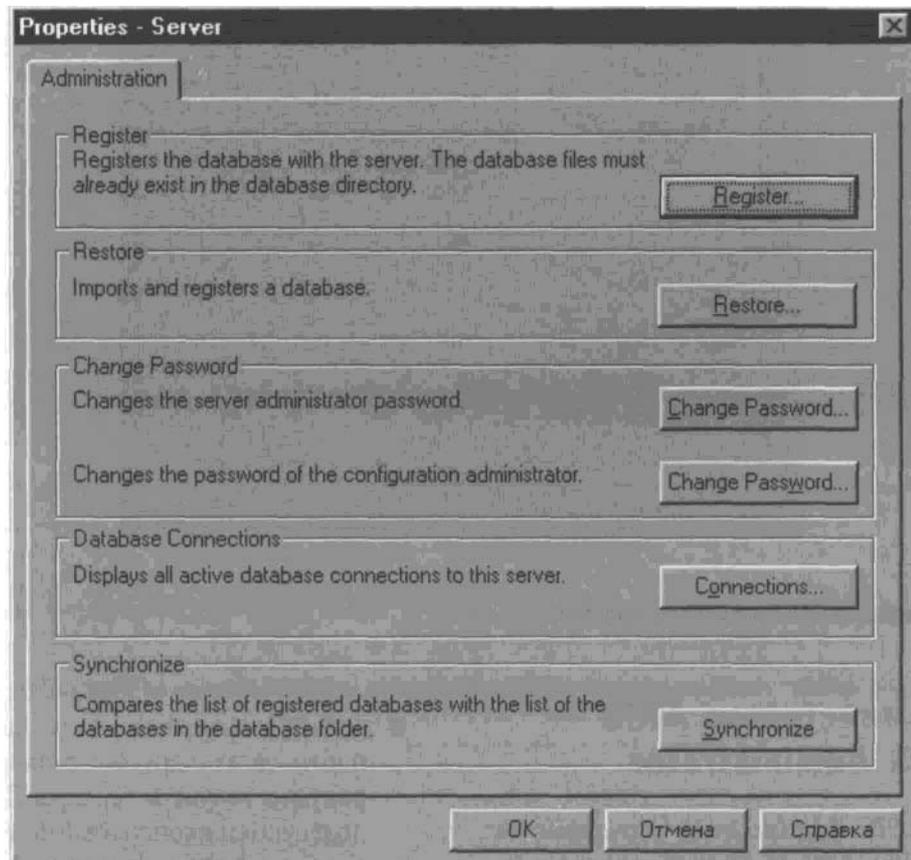


Рис. 73. Закладка Administration окна Properties-Server (Свойства сервера)

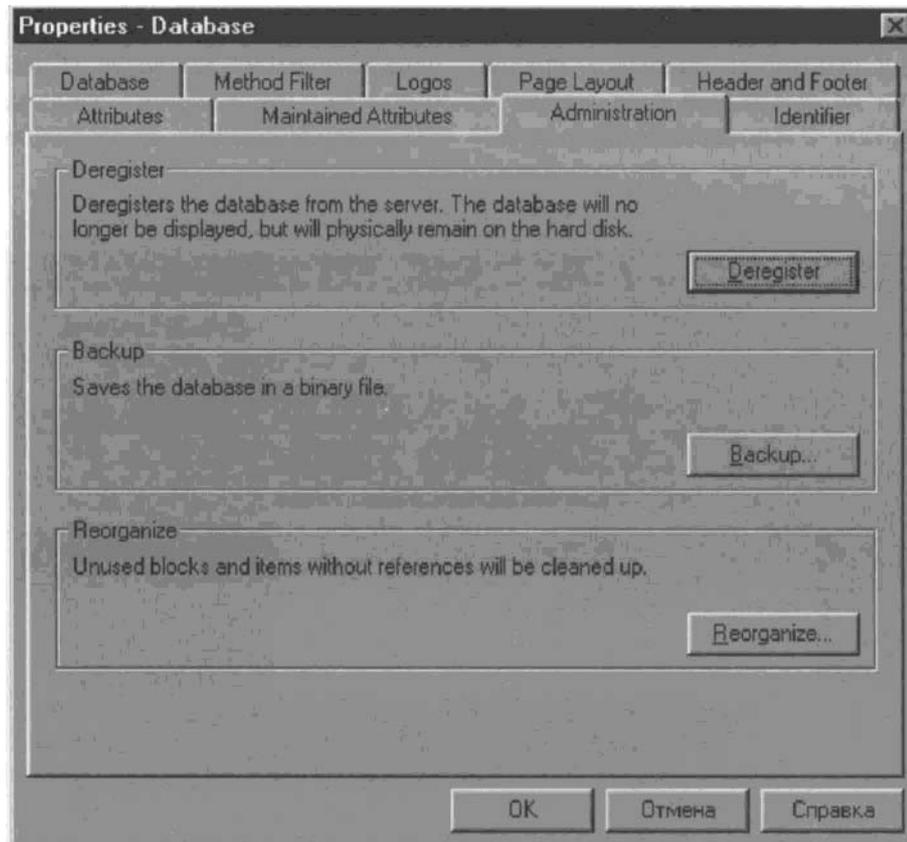


Рис. 74. Закладка Administration окна Properties Database (Свойства базы данных)

## элементов ARIS — APIS Identification

### 5.12. Идентификация

Модуль *ARIS Identification* предназначен для создания уникальных идентификаторов элементов базы данных. Идентификатор — алфавитно-цифровая строка для однозначного определения элемента базы данных. Он состоит из префикса и последовательного номера. Префикс и последовательный номер отделены друг от друга десятичной точкой. Идентификатор элемента входит в его атрибуты и может быть просмотрен в окне *Attributes* (рис. 30).

Идентификаторы не зависят от используемых в базе данных языков, что является преимуществом при поиске элементов, в атрибутах которых применяются различные языки.

Модуль *ARIS Identification* выполняет присвоение, изменение и удаление идентификаторов, а также префиксное управление. Идентификация может производиться автоматически (при создании элементов базы данных) или вручную позднее по времени.

Идентификаторы присваиваются следующим элементам базы данных:

- объектам;
- связям;
- моделям;
- таблицам;
- пользователям и их группам;
- языкам;
- форматам шрифтов.

Для включения режима автоматической идентификации необходимо щелкнуть правой кнопкой мыши на выбранной базе данных в окне ARIS Explorer. В контекстном меню вы-

бирается пункт меню *Properties*. В появившемся окне *Properties Databases (Свойства базы данных)* открывается закладка *Identifier (Идентификатор)*, приведенная на рис. 75.

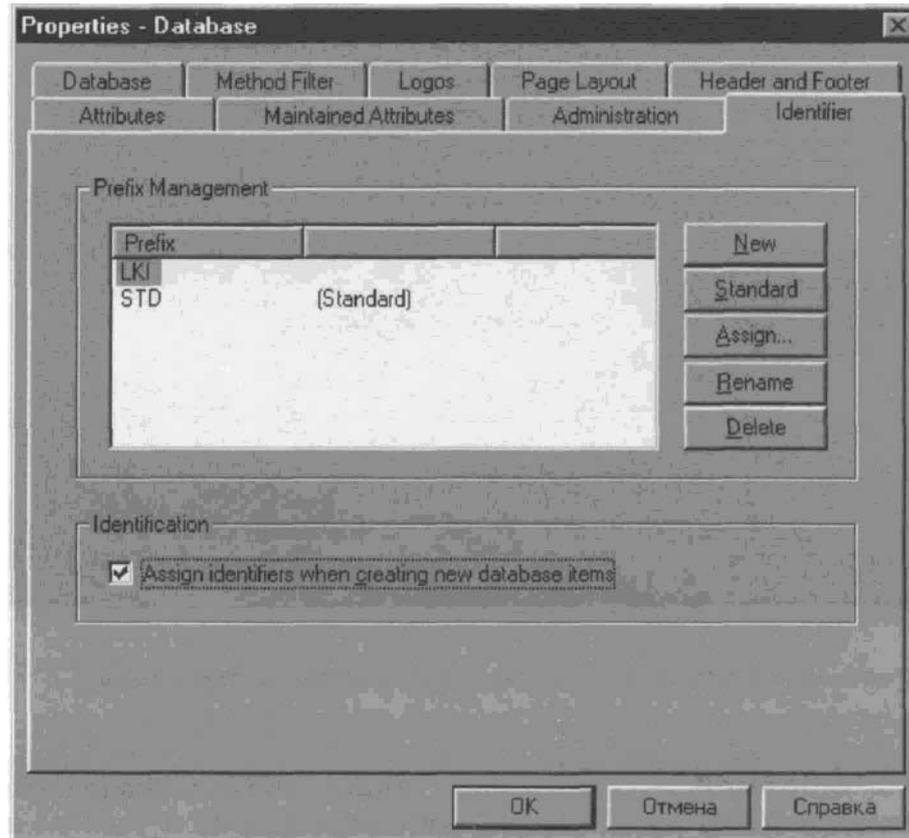


Рис. 75. Закладка Identifier

Она доступна только тогда, когда у пользователя есть доступ к префиксному управлению.

Для включения автоматической идентификации необходимо поставить флажок *Assign identifiers when creating new database items (Присваивать идентификатор, когда создается новый элемент базы данных)*. При создании новой базы данных автоматическая идентификация выключается по умолчанию.

ARIS использует стандартный префикс *STD* базы данных для автоматической идентификации до тех пор, пока пользователь не назначит другой префикс.

Привилегированный пользователь может назначить свой собственный префикс (кнопка *New* на рис. 75), а также и назначать его (кнопка *Assign*) его для определенных пользователей и их групп (рис. 76) путем расстановки флажков рядом с именами пользователей и названиями групп.

Идентификация, выполняемая в ручную позволяет назначать идентификаторы по прошествии времени после создания элемента, а также изменять или удалять существующие идентификаторы.

## Инструментальная система ARIS

Механизм управления идентификаторами реализуется в двух последовательно открываемых окнах *Identifier Wizard* (*Помощник по управлению идентификаторами*) (рис. 77 - 78). Первое окно вызывается через пункт *Assign Identifiers* контекстного меню. В начале предлагается сделать выбор между назначе

нием (*Assign Identifiers*) и удалением (*Delete Identifiers*) идентификатора. Причем в первом случае можно переписать существующий идентификатор, поставив флажок *Overwrite Existing Identifiers*.

Во втором окне нужно выбрать префикс, который будет использоваться при идентификации (рис. 78).

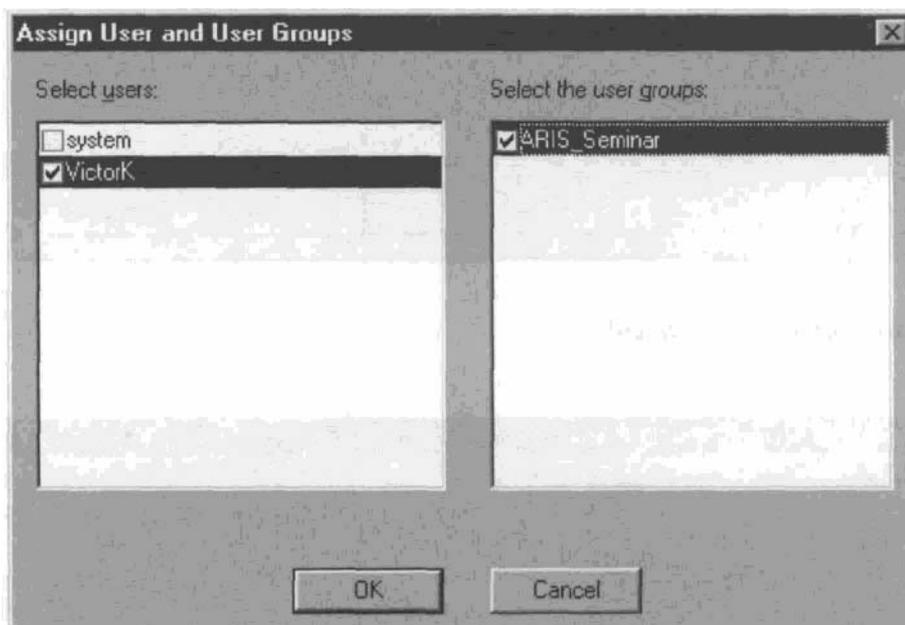


Рис. 76. Закрепление префиксов за пользователями и их группами



Рис. 77. Первое окно Identifier Wizard



Рис. 78. Второе окно Identifier Wizard

## 5.13. Конфигурирование системы — ARIS Configuration

Модуль *ARIS Configuration* позволяет корректировать методы и объекты ARIS, приспособив их к требованиям конкретных пользователей или специфических задач. Для его применения пользователь должен иметь право администратора конфигурации.

С помощью *ARIS Configuration* пользователь может переопределить:

- модели;
- объекты;
- соединения;
- атрибуты;
- детализацию объектов;
- символы объектов;
- формат шрифтов.

В папке *Configuration* (рис. 65) сетевого окружения ARIS находятся инструменты *ARIS Configuration*:

- фильтры конфигурации (*Configuration Filter*), включающие методологические фильтры (*Method Filter*) и фильтры оценки (*Evaluation Filter*);
- шаблоны (*Templates*).

Для каждого сервера имеется одна база данных конфигурации (папка *Configuration*), содержание которой будет использоваться всеми базами данных сервера.

Фильтры конфигурации позволяют оптимально адаптировать модели к различным задачам и требованиям пользователей, обеспечивая тем самым ориентацию на определенные рекомендации по моделированию. Кроме того, они позволяют дозировать предоставляемую пользователям информацию.

Фильтры конфигурации идентифицируются ключом фильтра, названием и иногда — соответствии с описанием. Название и описание могут быть на нескольких языках.

Методологические фильтры представляют собой регулируемые, переключаемые наборы моделей ARIS. Они являются эффективным средством поддержки однородного моделирования, способным обеспечивать пользователей ограниченным количеством необходимых методов моделирования. В ARIS определены стандартные фильтры, рассмотренные в разделе 4.3.

Фильтры оценки и методологические фильтры с формальной точки зрения идентичны. Однако они предназначены для решения разных задач. Методологические фильтры определяют набор моделей, которые может применять пользователь. Фильтры оценки могут использоваться во всех модулях ARIS, позволяющих получать оценки моделей. Они помогают пользователю в более точной спецификации информации, доступной в текущем методологическом фильтре.

Шаблоны дают возможность пользователю определять индивидуальные настройки параметров для элементов ARIS. Шаблон может применяться для любой модели в *ARIS Designer*. С помощью шаблонов достигается более индивидуальное проектирование моделей.

Основу *ARIS Configuration* составляют:

- механизм управления построением фильтра конфигурации (*Configuration Filter Wizard*). Он позволяет выбирать необходимые типы моделей, определять объекты, символы, типы соединения и их атрибуты, создавая тем самым методологические фильтры и фильтры оценки;
- механизм управления созданием шаблона (*Template Wizard*), при помощи которого можно определить изображение объектов и связей, а также включать эти шаблоны в отдельные модели.

Эти механизмы запускаются посредством контекстного меню, вызываемого на закладках *Filter* и *Templates* папки *Configuration*. Последовательно выполняя рекомендации, появляющиеся в окнах *Configuration Filter Wizard*, пользователь может построить нужные ему фильтры и шаблоны.

Функциональные возможности *AR/S Configuration* позволяют экспортировать и импортировать фильтры конфигурации и шаблоны.

Необходимо отметить, что модуль *ARIS Configuration* следует применять с большой осторожностью, так как это требует глубоких знаний методологии и всей инструментальной системы.

#### 5.14. Консолидация объектов баз данных — ARIS Consolidation

Модуль *ARIS Consolidation* предназначен для очищения используемых баз данных и устранения их избыточности. Он позволяет объединять ненужные двойные копии определений объектов одного типа в пределах базы данных в единственное определение объекта. Для работы с модулем *ARIS Consolidation*

пользователю необходим привилегированный доступ к базе данных.

Для упрощения процедуры консолидации в ARIS предусмотрен соответствующий механизм (*Consolidation Wizard*), который поддерживает пользователя на всех этапах. Для его запуска нужно сначала найти идентичные объекты. Для этого необходимо вызвать контекстное меню на базе данных и выбрать пункт *Find/Objects with Identical Names* (*Найти объекты с одинаковыми именами*). В открывшемся окне (рис. 79) нажимается кнопка *Start*, инициирующая поиск.

В списке найденных объектов выделяется нужный объект и вызывается контекстное меню, в котором выбирается пункт *Consolidate* (*Консолидировать*). После этого запускается механизм консолидации и появляется первое окно (рис. 80), где необходимо выбрать консолидируемые объекты. К имеющемуся списку могут быть добавлены другие (кнопка *Add*).

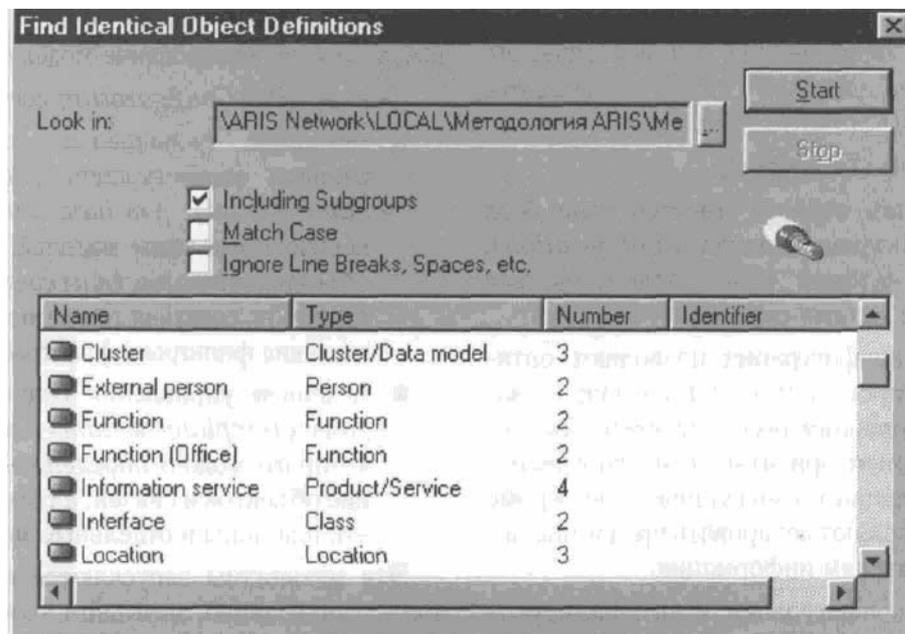


Рис. 79. Окно поиска объектов с одинаковыми именами

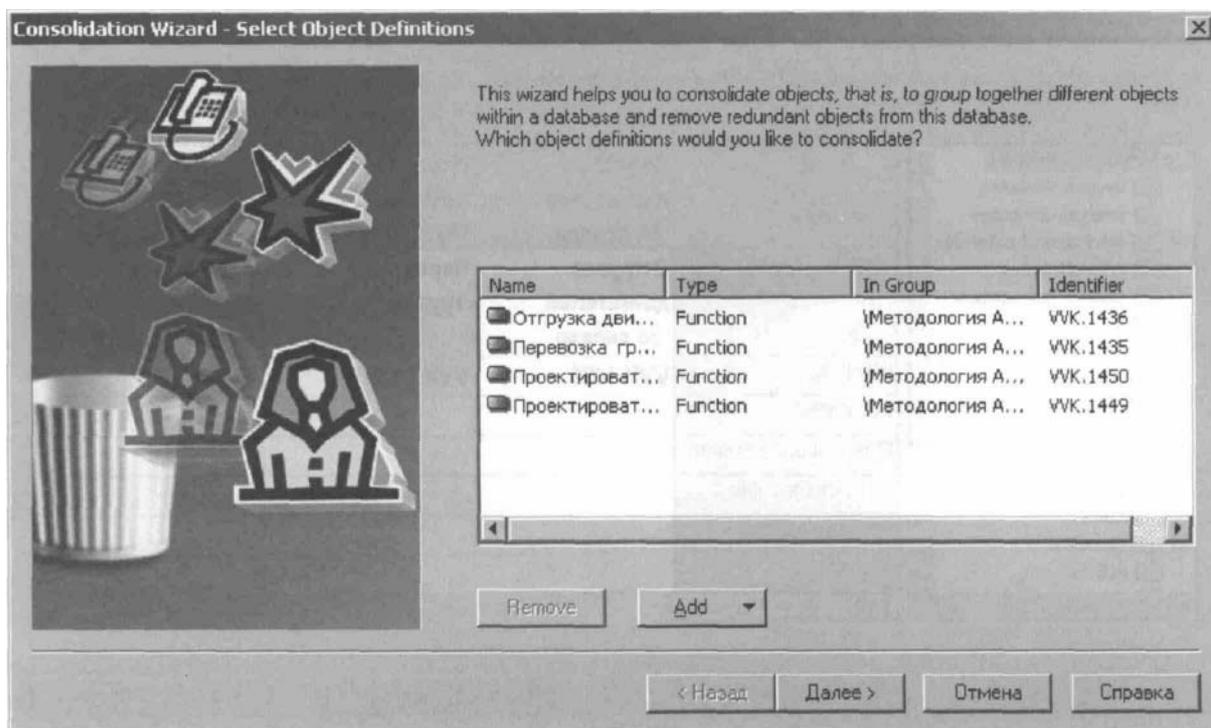


Рис. 80 Первое окно Consolidation Wizard. Выбор консолидирующих объектов

Во втором окне (рис. 81) выбирается главный объект *{Master object}*, с которым будут консолидироваться остальные объекты. Выбор производится нажатием мыши на столбец, содержащий атрибуты этого объекта. Его заголовок выделяется красным цветом, и становится активной кнопка *Master object*, которую нужно нажать для перехода к третьему, последнему, окну (рис.82).

В нем можно выбрать следующие опции:

- стереть избыточные определения объектов после консолидации (*Delete redundant object definition after consolidation*),
- соединить атрибуты (*Merge attributes*);
- не консолидировать множественные связи, если это разрешено методом (*Do not consolidate multiply-occurring relationships if permitted by the method*).

После нажатия кнопки *Готово* начинается процесс консолидации, информация о ходе которого отражается в выходном окне ARIS.

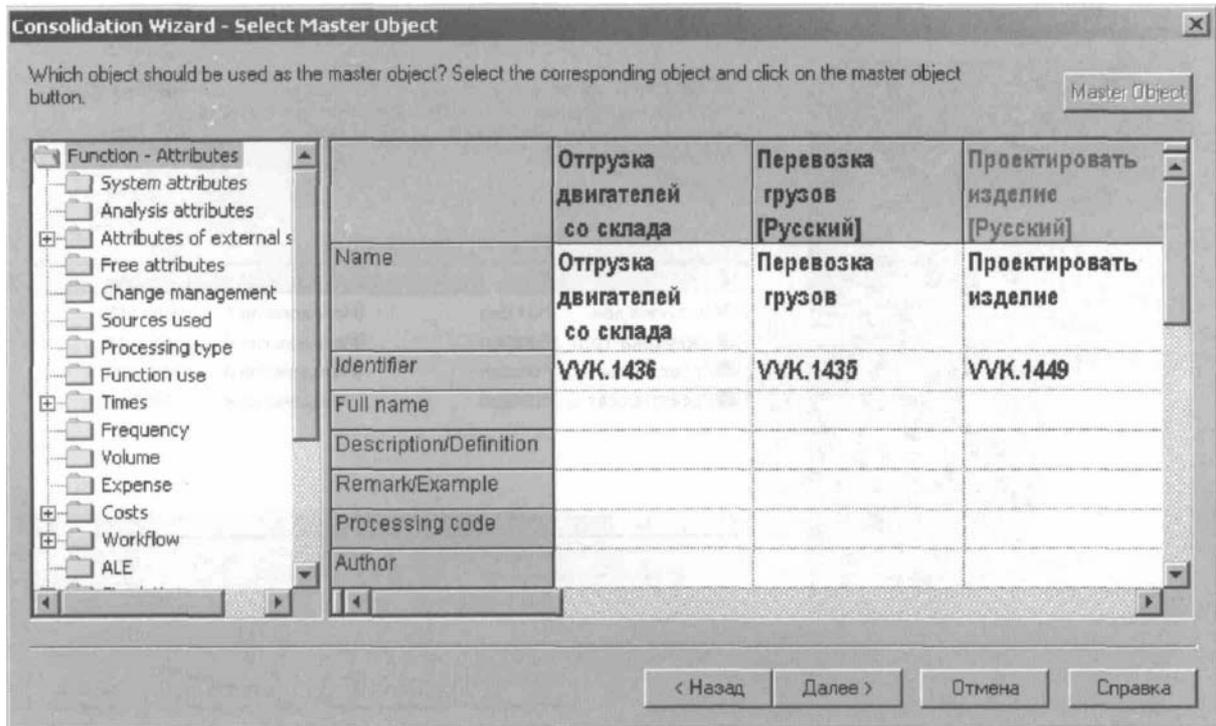


Рис. 81. Второе окно Consolidation Wizard. Выбор главного объекта

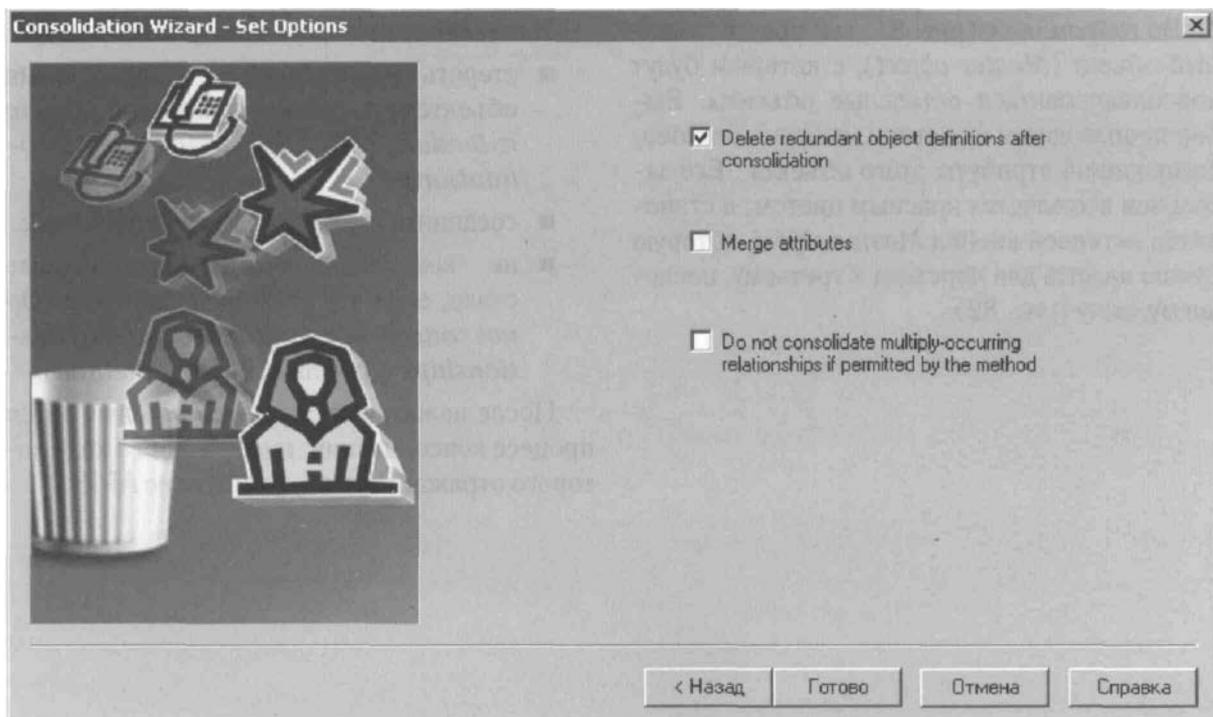


Рис. 82. Третье окно Consolidation Wizard. Задание опций консолидации

## 5.15. Объединение моделей — ARIS Merge

Модуль *AR/S Merge* позволяет копировать модели и объекты в одной базе данных. Это необходимо для объединения результатов работы нескольких независимых команд, а также для организации процесса накопления и использования прототипов проектных решений.

*ARIS Merge* входит в модуль *ARIS Toolset*. Он также доступен с ограниченными функциональными возможностями в *ARIS Easy Design*. Если для объединения моделей в одной базе данных используется модуль *ARIS Merge*, то гарантируется наиболее непротиворечивое и свободное от избыточности состояние базы данных. Перед объединением моделей базы данных, подготовленные в различных подразделениях организации, должны быть консолидированы с помощью модуля *ARIS Consolidation*.

При объединении содержания различных баз данных ARIS можно выбирать между объединением (*Merge*) и модифицированным объединением (*Update Merge*).

При объединении баз данных, созданных в ARIS 5.0, возможно объединение двух элементов с одинаковым идентификатором *GUID*. В отличие от баз данных ARIS 5.0 элементы версий 4.0 и предыдущих не имеют идентификатора *GUID*. Если при использовании ARIS 5.0 впервые открываются базы данных, созданные в ARIS 4.0, то идентификатор *GUID* будет назначен автоматически. Для объединения баз данных версий 4.0 и более ранних с базами версии 5.0 используется модифицированное объединение (*Update Merge*).

*ARIS Merge* имеет определенные преимущества по сравнению с возможностями модуля *ARIS Export/Import* и функцией *Save/Restore* модуля *ARIS Administrator*. Настройка процессов объединения проводится (рис. 83) с помощью закладки *Merge* (меню *View/Options*). ARIS дает возможность настраивать параметры в следующих группах или раскрывающихся списках:

- общие установки (*General*);
- атрибуты (*Attributes*);
- детализация (*Assignments*);
- разрешение конфликтов (*Conflict Resolution*) для моделей, объектов, таблиц, пользователей, групп пользователей.

В общих установках определяется вид объединения — простое (*Simple Merge*) или расширенное (*Extended Merge*).

Если комбинируются содержание баз данных, использующих все языки, доступные в исходной и целевой базах, то проводится простое объединение. Возникающие при этом конфликты могут быть решены во всех папках целевой базы данных согласно параметрам настройки в группе *Conflict Resolution*.

Если при объединении нужно выбрать доступные в базах языки, то нажимают кнопку выбора *Extended Merge*. Необходимые параметры настройки для расширенного объединения содержатся в диалоговом окне *Language Selection (Выбор языка)*.

В группе *Attributes* определяется будут ли (*Yes*) или нет (*No*) объединены атрибуты элемента-источника и элемента-приемника.

В текстовое поле *Assignment Level*, вводится число уровней детализации, которые должны рассматриваться при объединении баз.

В группе *Conflict Resolution* определяется порядок объединения для модели, объекта, таблицы, пользователя или группы пользователей. Процедура объединения запускается следующим образом:

- выбирается копируемый элемент (база данных, папки, пользователи, шрифты, модели, объекты);
- посредством обычного или контекстного меню производится копирование или вырезание выбранного элемента;
- затем выделяется база данных, с которой необходимо провести объединение;
- в ее контекстном меню выбирается один из пунктов *Paste (Вставить)*, *Paste as/Merge (Вставить как/Слияние)* или *Paste as/Update Merge (Вставить*

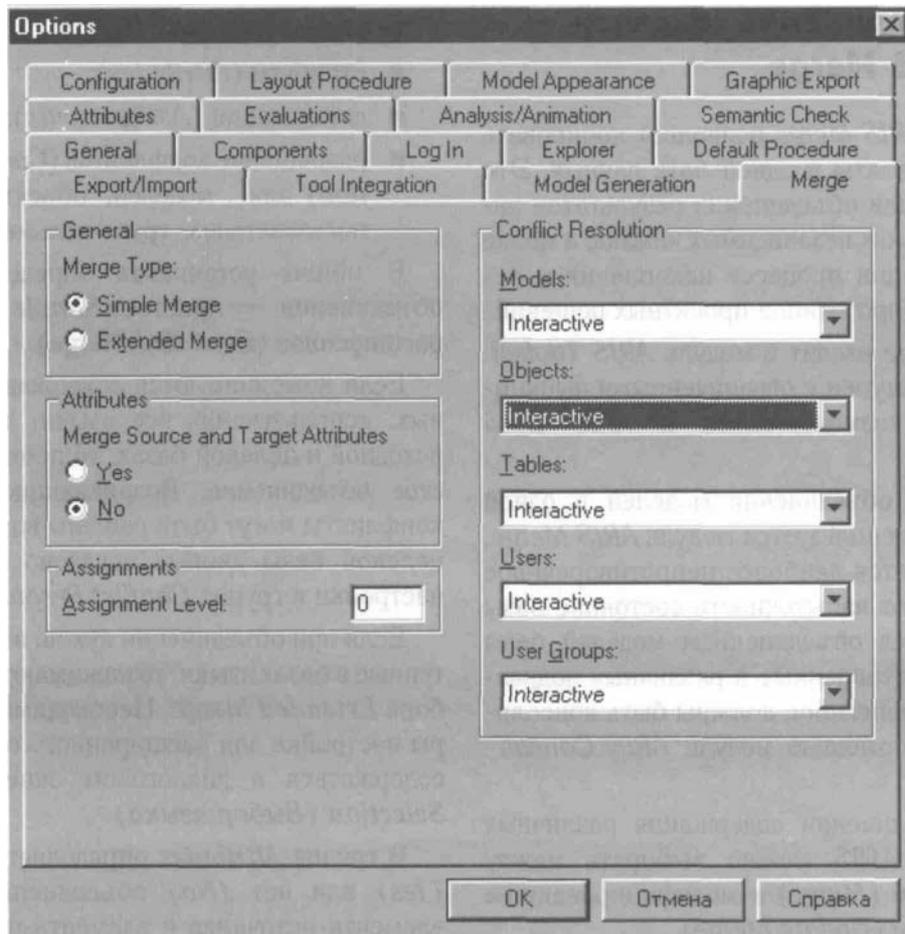


Рис. 83. Настройка опций ARIS Merge

как/Модифицировать объединение), и процесс объединения начинается. Его протекание отражается в окне вывода ARIS.

Имеется возможность проводить объединение методом *Drag&Drop* (*Перетащить и оставить*).

## 5.16. Генератор моделей — ARIS Model Generator

Модуль *ARIS Model Generation* позволяет создавать новые виды моделей, используя существующие в базе данных модели и объекты, которые при этом остаются неизменными.

Модуль *ARIS Model Generation* в отличие от *ARIS Process Generator* не создает никаких новых объектов или типов связей. Создаются только дополнительные экземпляры объектов. Размещение новых моделей производится автоматически. Тексты, OLE-объекты и графические объекты не могут быть использованы при генерации модели.

Модуль *ARIS Model Generation* работает совместно с *ARIS Designer* и предоставляет рентабельные функциональные возможности для разработки новых моделей, сокращая время этого процесса.

В *ARIS Model Generation* реализованы две процедуры. Благодаря механизму генерации моделей *Model Generation Wizard* (*Помощник генерации моделей*) возможно:

## Инструментальная система ARIS

- создание новой модели на основе существующих (*Model Generation Based on Models*);

- «сборка» новой модели путем выбора отдельных существующих объектов (*Model Generation of Not-Lane Models*).

Например, можно создать дерево функций, используя функции, содержащиеся в нескольких диаграммах ЕРС.

Механизм *Model Generation Wizard* позволяет быстро создавать новые модели, с помощью которых можно проводить анализ бизнес-процессов с различных точек зрения, в том числе концентрируя внимание на подпроцессах.

Используя закладку *Model Generation (Генерация моделей)* в меню *View/Options* (рис. 84), можно задать некоторые параметры механизма генерации моделей.

При генерации новой модели на основе существующих возможен перенос из них как объектов и связей, так и только связей (флажок в поле *Transfer only relationships from the source models (Переносить одни связи из модели-источника)* на рис. 84).

При создании моделей на основе объектов (режим *Model Generation of Not-Lane Models*) возможен перенос в новую модель нескольких существующих объектов при установке флажка *Several occurrences of non-structurally-relevant objects (Несколько включений не структурно-образующих объектов)*, показанного на рис. 84. Например, если необходимо сгенерировать диаграмму eЕРС, в которой объект «организационная единица» исполняет несколько функций, то он будет воспроизведен в каждой функции. Это сделает модель более легкой для чтения.

В текстовом поле *Display Depth (Глубина показа)* можно определить глубину, до которой будут показаны связи между объектами. Максимальная глубина показа 5. При глубине, равной 1, будут представлены все объекты с прямыми отношениями к выбранному объекту.

Механизм генерации моделей (*Model Generation Wizard*) можно запустить через пункт *Generate Model (Генерировать модель)*

контекстного меню при выбранной в *ARIS Explorer* модели или объекте. Можно также запустить *ARIS Model Generation*. Для этого нужно выбрать в *ARIS Explorer* пункт меню *Edit/Find (Редактировать/Найти)* для поиска требуемых моделей. После проведения поиска следует выбрать один или несколько найденных элементов и открыть контекстное меню правой кнопкой мыши, пункт *Generate Model*.

Механизм генерации моделей *Model Generation Wizard* предполагает несколько этапов, в течение которых выбираются:

- исходные модели или объекты (рис. 85);
- новый тип модели (рис. 86);
- папка и имя модели (рис. 87).

После ввода всех необходимых параметров новая модель будет сгенерирована, автоматически сохранена и открыта для рассмотрения.

## 5.17. Варианты моделей — ARIS Variants

Модуль *ARIS Variants* предназначен для создания вариантов одной и той же модели, соответствующих различным начальным условиям и состояниям моделируемого объекта. Варианты моделей, учитывающие разные условия, дают возможность проводить анализ с различных точек зрения. Удобным может оказаться сравнение исходной модели (*Master model*) и ее варианта (*Variant model*) по нескольким критериям.

Модуль *ARIS Variants* позволяет также головной организации на основе описанных ее основных процессов строить варианты в филиалах с учетом их специфики.

Создание вариантов и сравнение элементов осуществляются при помощи механизма *Variant Wizard (Помощник создания вариантов)*. В первом случае этот механизм запускается посредством пункта *Create Variants (Создать вариант)* контекстного меню выбранной модели-мастера.

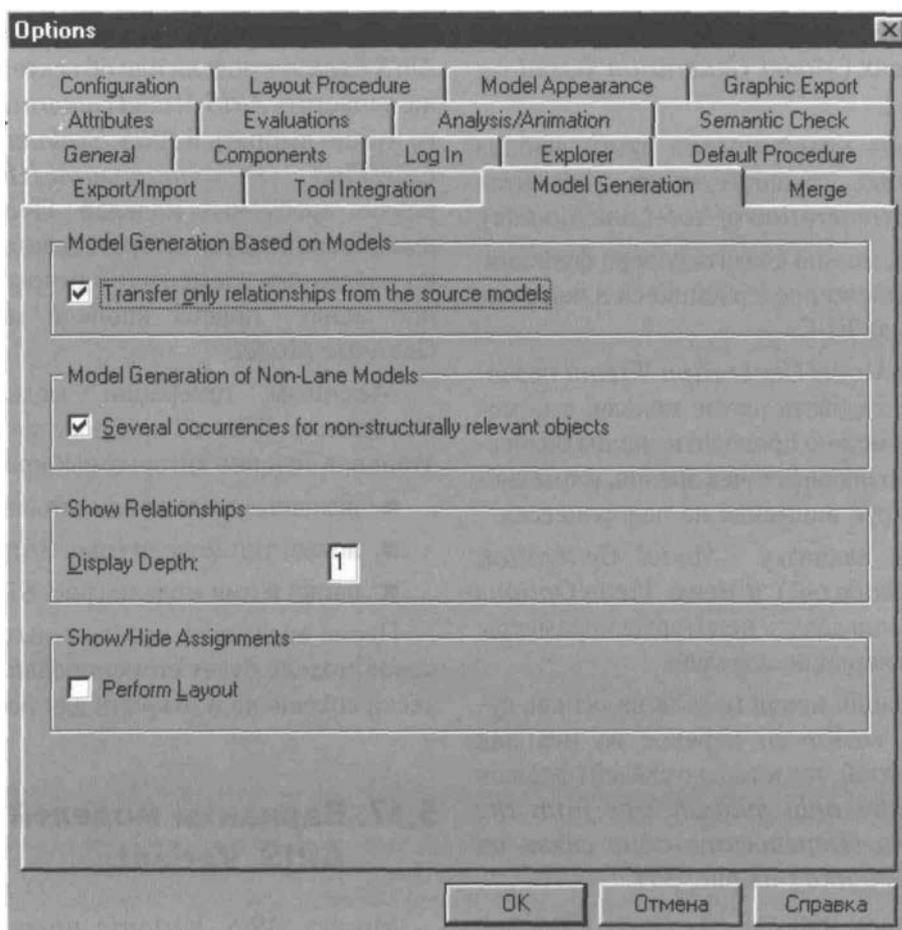


Рис. 84. Параметры генерации моделей в модуле ARIS Model Generation

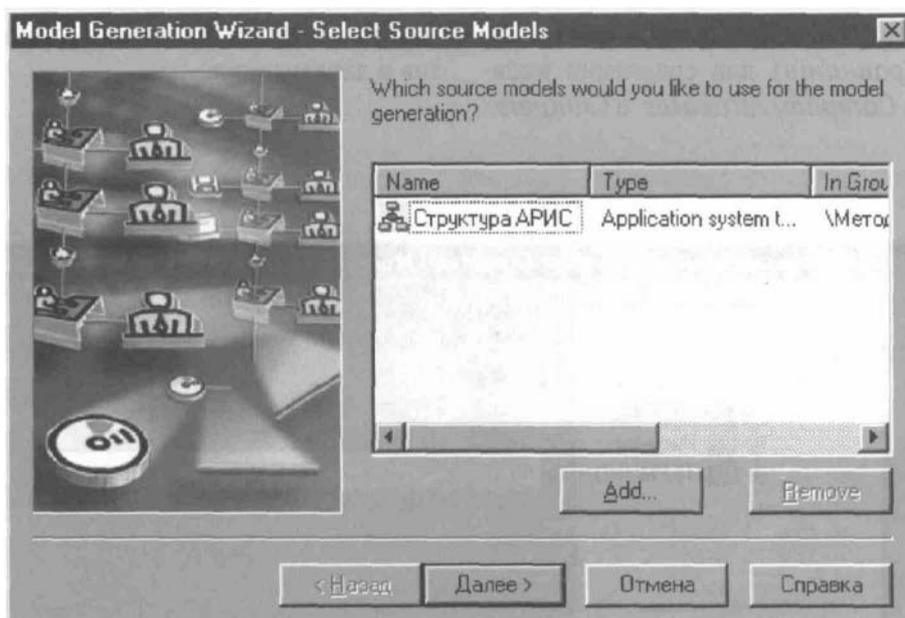


Рис. 85. Окно Model Generation Wizard. Выбор исходных моделей

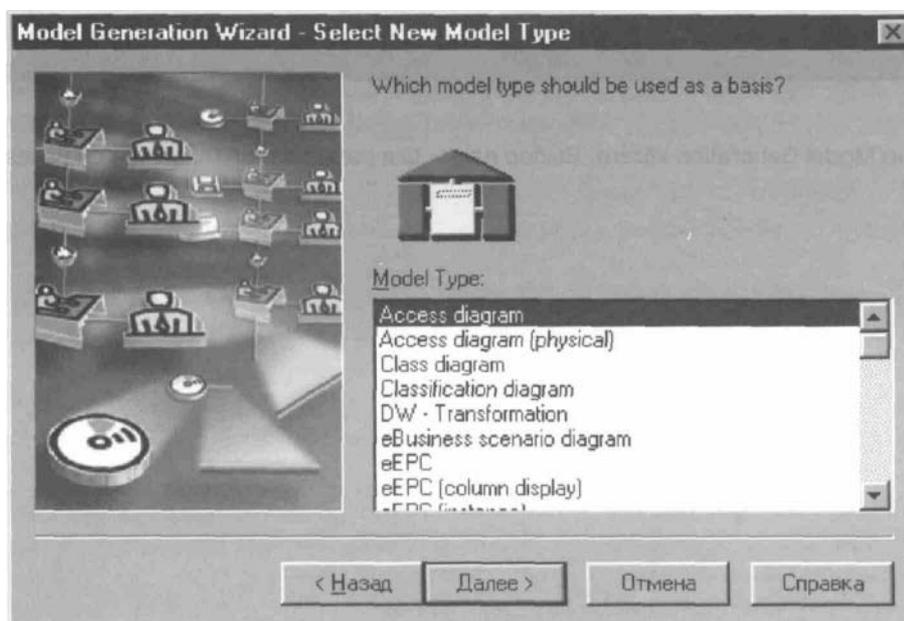


Рис. 86. Окно Model Generation Wizard. Выбор типа генерируемой модели

**Моделирование бизнеса. Методология ARIS**

Во втором случае используются пункт меню *Compare* (*Сравнить*) для сравнения моделей и пункты *Compare/Attributes* и *Compare/*

*Assignments* для сравнения атрибутов объектов и детализаций.

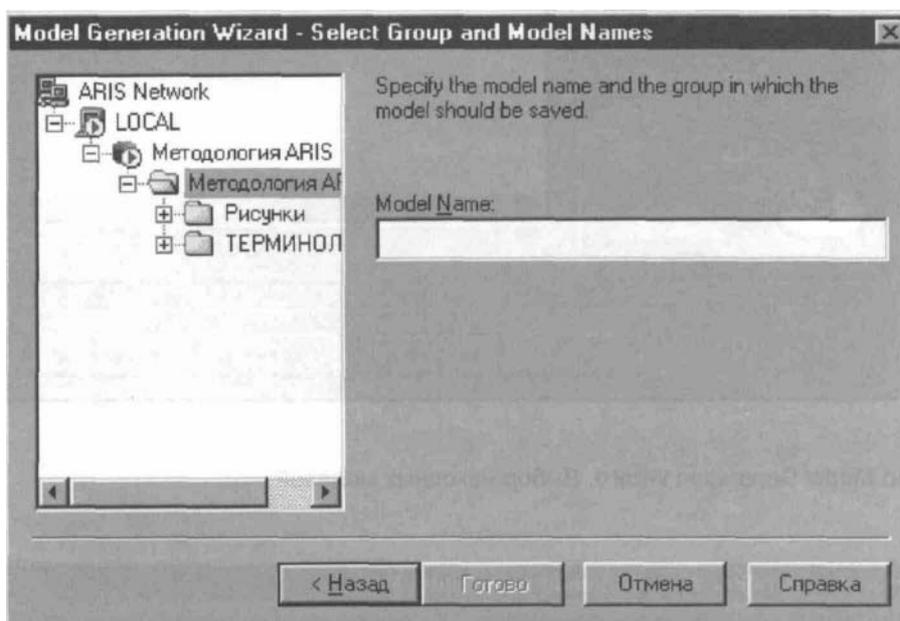


Рис. 87. Окно Model Generation Wizard. Выбор папки для размещения модели и ее названия

## Инструментальная система ARIS

## 5.18. Экспорт и импорт элементов — ARIS Export/Import

Модуль *ARIS Export/Import* предназначен для обмена содержанием баз данных с программными средствами других производителей. Если необходимо передать содержание одной базы данных ARIS другой базе ARIS, то лучше использовать модули *ARIS Merge* или *ARIS Administrator*.

Главное назначение модуля *ARIS Export/Import* — обмен отдельными моделями с другими программными CASE-системами посредством файлов, записанных в ASCII-

кодах. Этот обмен осуществляется при помощи специальных программ-интерфейсов сторонних производителей, например компании *Reischmann Informatik GmbH (RI)*, которая специализируется в разработке интерфейсов для инструментальных CASE-средств и репозитариев (см. табл. 2). В настоящее время интерфейсы TOOLBUS компании RI поддерживают более 30 инструментальных CASE-средств. Во всем мире более 100 компаний используют интерфейсы TOOLBUS. Компания RI является партнером компании Rational Software for Rational Rose.

Таблица 2. Интерфейсы TOOLBUS

Название программного продукта, версии	Экспортируемые/импортируемые модели	Версия TOOLBUS	Версии ARIS
ERWin, версии 2.6, 3.0, 3.5 и 3.5.2	Модели eERM, SAP-SERM, модель данных IEF, диаграмма атрибута	5	3.0,3.1, 3.2, 4.0, 4.1
CASE /4/0, версии 4.3a/4.3b	Модели eERM, SAP-SERM, модель данных IEF, диаграмма атрибута, дерево функции	1	4.0,4.1
COOL: BIZ, версии 4.0, 4.1, 5.0 и 5.1	Модель данных IEF, модели eERM, SAP-ERM, диаграмма атрибута, дерево функций, диаграмма окружения	2	3.0,3.1, 3.2, 4.0, 4.1
COOL: BusinessTeam, версии 1.2 и 1.2a, Bachman /AnalystReleases 4.15, 4.20, 4.30 и 4.4, Groundworks Releases 1.x и 2.x	Модели eERM, SAP-SERM, модель данных IEF, диаграмма атрибута, дерево функции	4	3.0,3.1, 3.2, 4.0, 4.1
COOL: ENTERPRISE (KEY/ADW) Версия ADW/KEY 2.7 (и выше) и COOL: Enterprise версия 4.0 (и выше)	Модели eERM, SAP-SERM, модель данных IEF, диаграмма атрибутов, дерево функций, диаграмма окружения функции	5	3.0,3.1, 3.2, 4.0, 4.1
COOL: GEN, версии 4.1a, 5.0и5.1	Модели eERM, SAP-SERM, модель данных IEF, диаграмма атрибута, дерево функций, диаграмма окружения функции	3	3.1,3.2, 4.0,4.1

Продолжение табл. 2

Название программного продукта, версии	Экспортируемые/импортируемые модели	Версия TOOLBUS	Версии ARIS
Oracle Designer, версии 1.2, 1.3, 2.1, 2.1.2 и 6.0	Экспорт и импорт: модели eERM, SAP-SERM, модель данных IEF, диаграмма атрибута, дерево функции, диаграмма окружения функции. Экспорт из ARIS: модель eEPC, организационная схема	7	3.0,3.1, 3.2,4.0,4.1
PowerDesigner/ DataArchitect, версии 6.0 и 6.1	Модели eERM, SAP-SERM, модель данных IEF, диаграмма атрибута	3	3.0,3.1, 3.2,4.0,4.1
Rational Rose, версии 98i, 2000 и 2000e	UML-диаграмма класса, UML-диаграмма описания класса, UML-диаграмма использования приложения, UML-диаграмма компонентов, UML-диаграмма действия	5	4.0,4.1 и 5.0
SELECT Enterprise, версия 6.1	UML-диаграммы класса, UML-диаграммы использования приложения, модель данных IEF, модели eERM, SAP-SERM, диаграмма атрибута, дерево функций	3	4.0,4.1
SELECT SE (Systems Engineer), версии 6.x и 7.0	Модели eERM, SAP-SERM, модель данных IEF, диаграммы атрибутов, дерево функций, диаграмма окружения функций.	2	3.2,4.0,4.1
System Architect 2001	Модели eERM, SAP-SERM, диаграмма атрибута, дерево функций	1	3.2,4.0,4.1

Кроме приведенных в табл. 2 существуют интерфейсы с некоторыми другими программными продуктами, например, Fabasoft Components, Filenet Visual Workflow, Multidesk Access, Staffware Workflow, CSE, Designer/ 2000. Для связи с отдельными программными системами в ARIS включены специальные модули, такие как *ARIS Connectivity for Lotus Notes* и *ARIS Connectivity for R/3*.

Механизм передачи моделей из ARIS в другую программную систему и обратно состоит в следующем. При помощи модуля *ARIS Export/Import* выбранные модели ARIS преобразуются в экспортируемый файл, записанный в ASCII-формате. Этот файл при помощи

интерфейсов TOOLBUS преобразуется в другой файл, «понимаемый» конкретным программным продуктом. Обратный процесс происходит аналогичным образом.

Настройки процесса экспорта/импорта осуществляются на закладке *Export/Import* (рис. 88) окна *Options*, вызываемого через пункт меню *View/Options*.

Обмен данными управляется последовательностью окон *Export (Import) Wizard (Помощник экспорта (импорта))*, которые вызываются из пункта *Export/Import* контекстного меню, активизируемого на нужной модели. Можно также воспользоваться пунктом меню *File/New*. В результате появится окно

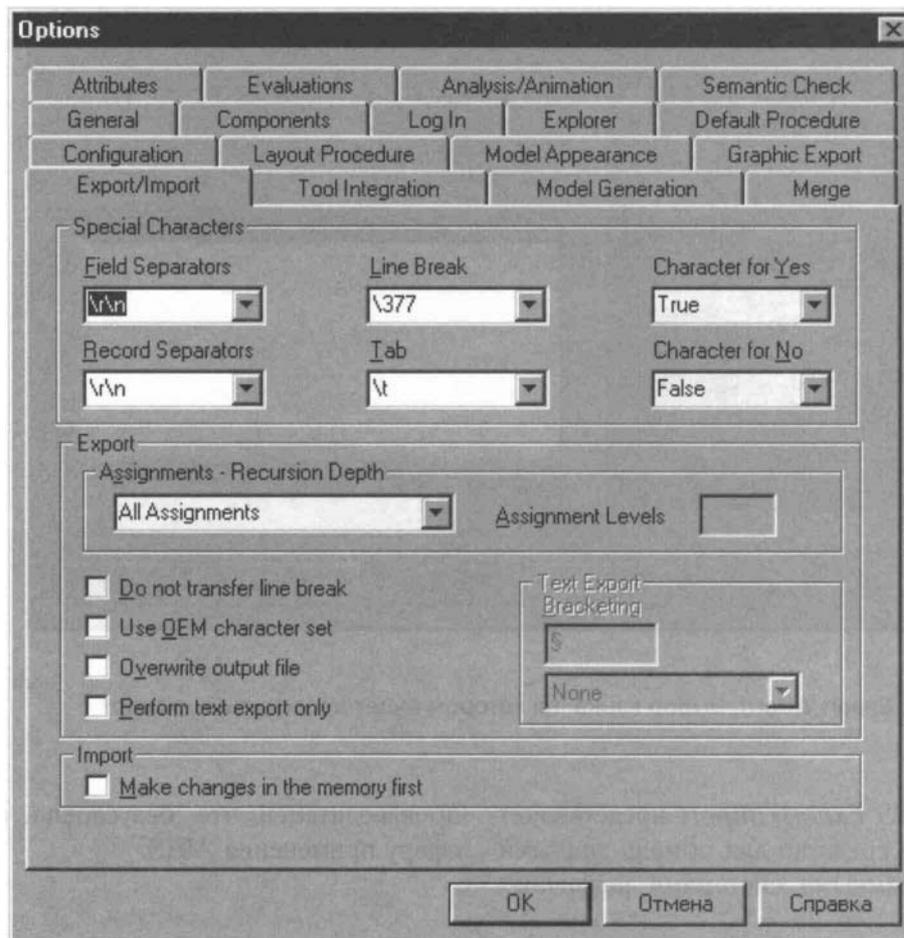


Рис. 88. Настройки процесса экспорта/импорта

(рис. 92), где в списке необходимо выбрать строку *Export File (Экспортируемый файл)*, а затем нажать кнопку *OK*. Появится окно, в котором нужно выбрать модель, информация о которой будет экспортироваться.

Далее пользователю предлагается выбрать язык, на котором будет проводиться экспорт информации (рис. 89). Затем пользователь должен указать файл, предназначенный для вывода информации (рис. 90) о выбранном элементе.

Созданный файл можно при желании просмотреть обычными средствами Windows.

При импорте (рис. 91) нужно указать файл, из которого будут импортироваться данные. Естественно, этот файл заранее должен быть создан интерфейсом TOOLBUS после экспорта из стороннего программного средства или модулем *ARIS Export/Import* в процессе экспорта из одной базы данных в другую.

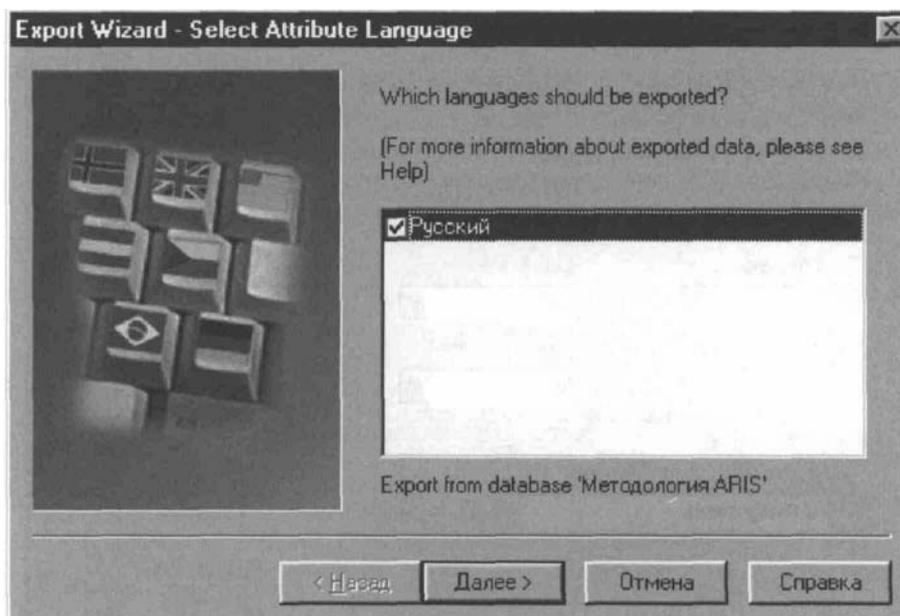


Рис. 89. Окно Export Wizard. Выбор языка, на котором будет проводиться экспорт

Модуль *ARIS Export/Import* представляет собой мощное средство для обмена данными между программными системами различных производителей, что, безусловно, расширяет сферу применения ARIS.

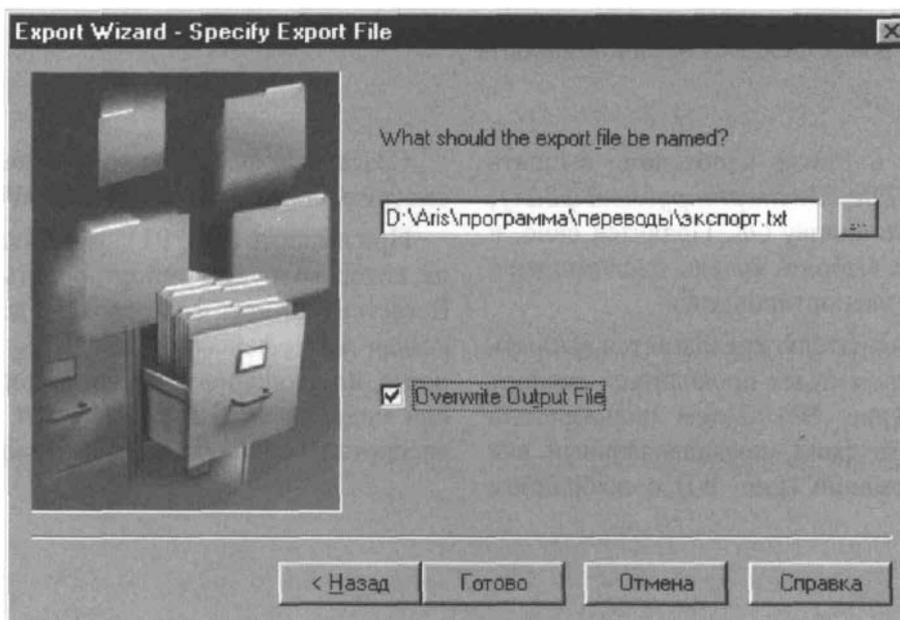


Рис. 90. Окно Export Wizard. Выбор файла для вывода экспортируемой информации

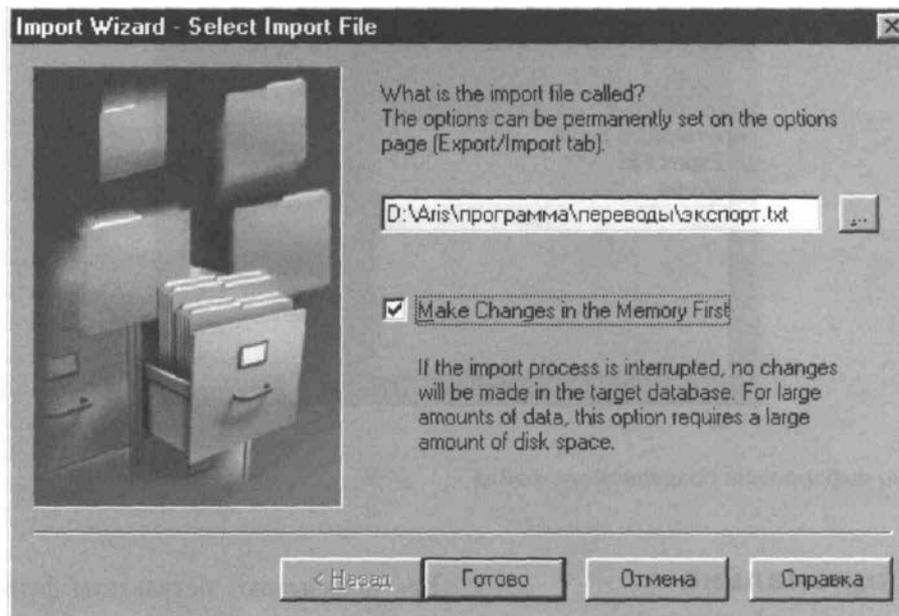


Рис. 91. Окно Export Wizard. Выбор файла, из которого будут импортироваться данные

### 5.19. Текстовый редактор — ARIS RTF Editor

Модуль *ARIS RTF Editor* дает возможность создавать и редактировать файлы в формате RTF непосредственно в ARIS. Он доступен как в модуле *ARIS Easy Design*, так и *ARIS Toolset*.

Модуль *ARIS RTF Editor* обеспечивает пользователя всеми функциями, необходимыми для редактирования текстовых документов, которые были созданы как отчеты модулей *ARIS Report*, *ARIS Analysis* или *ARIS Semantic Check*.

Редактор вызывается через пункт меню *File/New*. После его нажатия, появляется окно (рис. 92), где в списке необходимо выбрать строку *RTFDocument*, а затем нажать кнопку *OK*. После этого раскроется окно редактора.

Пункты меню и панель инструментов модуля *ARIS RTF Editor* — традиционные для текстовых редакторов.

### 5.20. Редактор скриптов — ARIS Script Editor

Модуль *ARIS Script Editor* входит в *AR/S Toolset* и позволяет пользователю создавать собственные скрипты отчетов, соответствующие потребностям компании, в дополнение к скриптам, включенным в пакет ARIS.

Скрипт — это инструкция, которая содержит команды языков Visual Basic и ARIS Basic® Language Capability. Компоненты в ARIS обращаются к скриптам, чтобы обеспечивать пользователя базы данных доступной информацией в форме таблицы или текста. Используя скрипты, созданные мастером скриптов, можно генерировать вывод информации в файлы форматов RTF, HTML и TXT.

Создание отчетов и типы скриптов рассмотрены в разделе 10.

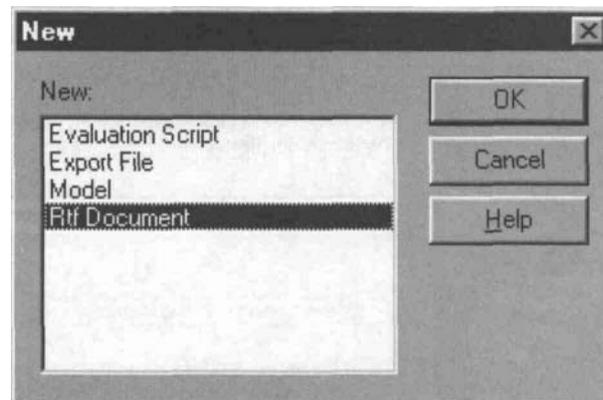


Рис. 92. Окно выбора типа создаваемого файла

## анализ — APIS ABC

### 5.21. Функционально-стоимостной

Модуль *ARIS ABC (Activity Based Costing)* интегрируется в базовый модуль *ARIS Toolset* и предназначен для проведения функционально-стоимостного анализа моделей. Его использование требует отдельной лицензии.

*ARIS ABC* реализует полноценный анализ стоимости процессов. Применение методов функционально-стоимостного анализа объясняется недостатками традиционных методов определения затрат, а также изменениями структуры затрат на предприятиях. Это связано в первую очередь с увеличением числа операций по подготовке, планированию, контролю и отслеживанию производственного процесса. Кроме того, усилиями руководителей производственные затраты сокращаются и не могут служить достоверным базисом для определения накладных расходов на производство продукции (товаров или услуг).

При традиционной системе расчета накладные расходы на производство отдельных товаров и услуг распределялись с использованием утвержденных коэффициентов, что не всегда точно отражало действительность.

Как показывают результаты функционально-стоимостного анализа, распределение накладных расходов между отдельными продуктами должен осуществляться на базе необходимых для их производства процессов. Процессы подразумевают всю последовательность действий по планированию, производству и реализации продукции конечному потребителю. Иными словами, накладные расходы перераспределяются от центров их возникновения (функциональных подразделений) к процессам, направленным на производство продукции.

К преимуществам применения методов функционально-стоимостного анализа можно отнести:

- прозрачность структуры накладных расходов;
- возможность более точного определения себестоимости продукции;
- выявление неэффективно используемых ресурсов;
- возможность проведения анализа типа «что, если?», т. е. возможность определить, что произойдет с затратами, если будут изменены процессы;
- обеспечение основы для периодического контроля процессов.

## Инструментальная система ARIS

Проведение функционально-стоимостного анализа в модуле *ARIS ABC* осуществляется для процессов, разработанных средствами семейства ARIS (*ARIS Toolset*, *ARIS Easy Design*, *ARIS Connectivity for R/3*). Единая интегрированная среда моделирования и анализа значительно повышает эффективность проведения подобного рода проектов. Возможность коллективной работы обеспечивает рациональное распределение обязанностей между специалистами — моделирование отдельных элементов происходит в каждом функциональном подразделении, а комплексный анализ всех бизнес-процессов проводится в отделе развития.

### 5.22. Интеграция с прикладными информационными системами — ARIS Tool Integration

Модуль *ARIS Tool Integration* позволяет обмениваться информацией о моделируемых процессах между *ARIS Toolset* и приложениями, разработанными другими изготовителями. Эти приложения, работающие вместе с *ARIS Toolset*, называются партнерскими.

*ARIS Tool Integration* делает доступным все функции, позволяющие интегрировать в ARIS системы партнера, конфигурировать передачу баз данных и проверять передаваемые данные.

*ARIS Tool Integration* — это модуль *ARIS Toolset*, требующий для установки дополнительной лицензии и инсталлируемый отдельно. После инсталляции он запускается путем вызова контекстного меню на выбранном элементе ARIS.

Модуль *ARIS Tool Integration* позволяет:

- конфигурировать *ARIS Toolset* так, чтобы можно было передавать информацию из базы данных ARIS в базы данных системы партнера;

- выполнять семантические проверки на системах партнера для информации, передаваемой из базы данных ARIS в базы данных системы партнера.

Для выполнения этих задач *ARIS Tool Integration* использует различные типы скриптов, которые могут быть созданы или отредактированы.

Для интеграции систем партнера *ARIS Tool Integration* использует три различных типа скриптов:

- скрипт соединения (*Coupling Scripts*);
- скрипт проверки (*Check Scripts*);
- скрипт семантических проверок (*SemCheck Scripts*).

Эти скрипты могут быть созданы и отредактированы с помощью редактора скриптов (*ARIS Script Editor*).

### 5.23. Преобразование баз данных ARIS 3.x в базы данных ARIS 5.0 — ARIS Database Converter

Модуль *ARIS Database Converter* преобразовывает базы данных ARIS 3.x в базы данных формата ARIS 5.0. Встроенный механизм преобразования (*Database Conversion Wizard* (Помощник конвертирования баз данных.)) поможет провести процесс преобразования формата.

Следует отметить, что этот процесс может продолжаться несколько часов для баз данных размером порядка 100 Mb.

### 5.24. ARIS и Интернет — ARIS Web Publisher

Модуль *ARIS Web Publisher* позволяет преобразовывать модели и объекты ARIS в формат HTML и отображать их в браузере. Использование этого модуля требует наличия специальной лицензии

Экспортируемые файлы формата HTML можно отображать при помощи браузеров Microsoft Internet Explorer и Netscape Communicator и делать эти файлы доступными для других пользователей через Intranet или

#### **Моделирование бизнеса. Методология ARIS**

Internet. Таким образом, устраняются препятствия для работы с моделями ARIS на разных территориях организаций.

Работу с ARIS Web Publisher существенно облегчает Web Publisher Wizard (Помощник публикации web-страниц).

## 6. Простой методологический фильтр. Обзор моделей

### 6.1. Организационные модели

#### 6.1.1. Организационная схема— Organizational chart

Организационная структура наиболее легко поддается моделированию, поскольку она в большинстве случаев четко определена и понятна. Рассматривая ее, следует различать организацию структуры и организацию процедур компании. Организационная структура включает правила, позволяющие произвести статичное структурирование подразделений компании. Организация процедур основана на правилах, отвечающих задачам, которые должны выполняться. Структура, ориентированная на задачи (в смысле распределения функций между исполнителями), в архитектуре ARIS представляется с помощью моделей управления. Таким образом, организационная схема является компонентой, которая позволяет анализировать организационную структуру компании.

Анализ организационной структуры должен проводиться с целью выявления:

- обоснованного количества уровней иерархии;
- наличия более чем 5-6 подчиненных подразделений у одного руководителя;
- наличия малого количества подчиненных у одного руководителя;
- подчинения одних и тех же звеньев (сотрудников) различным руководителям.

#### Простой методологический фильтр. Обзор моделей

Для анализа организационной структуры следует использовать схемы организационной структуры, представленные в произвольном формате, а также положения об отделениях, отделах, группах и т.д. На основе этой информации формируется организационная структура в виде организационной схемы ARIS.

Организационная схема описывает организационные единицы разного уровня и их взаимосвязи. Эта модель — одна из важнейших, так как она описывает субъекты, которые определяют выходы и входы потоков ресурсов предприятия.

В модели организационной структуры целесообразно показывать:

- подразделения предприятия;
- наименование должности и фамилии руководителей подразделений;
- физическое местоположение отделов на предприятии.

Моделирование организационной структуры — стартовая точка в создании топологии компьютерной прикладной системы, которая, как предполагается, будет поддерживать организационную структуру наиболее оптимальным образом. Соединения сети и сетевые узлы, расположенные в определенных местах компании, являются главными элементами топологии сети прикладной системы.

Организационная схема содержит 20 различных объектов, названия и графические изображения которых сведены в табл. 3, а возможные связи между элементами — в табл. 4. На рис. 93 показана панель инструментов «Моделирование» модуля *ARIS Designer* для организационной схемы.

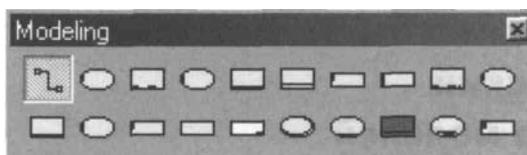


Рис. 93. Панель инструментов для моделирования организационных схем

Таблица 3. Объекты организационной схемы

	Английское название объекта	Русское название объекта
	Connection	Связь
	Organizational unit type	Тип организационной единицы
	Organizational unit type	Тип организационной единицы
	Organizational unit	Организационная единица
	Organizational unit	Организационная единица
	Cost center	Стоимостной фактор
	Position type	Тип сотрудника
	Position	Должность
	System organization unit type	Тип системной организационной единицы
	System organization unit type	Тип системной организационной единицы
	System organization unit	Системная организационная единица
	System organization unit	Системная организационная единица
	Person type	Тип сотрудника
	Internal person	Штатный сотрудник
	External person	Внештатный сотрудник
	Group	Группа
	Location	Расположение
	Organizational chart	Организационная схема
	Workstation	Рабочая станция
	Position description	Описание должности

Таблица 4. Связи объектов организационной схемы

Английское название связи	Русское название связи
Belongs to	Принадлежит
Can be constituent	Может являться частью
Can be disciplinary superior	Может быть непосредственным руководителем
Can be technical superior	Может быть техническим руководителем
Can Belongs to	Может принадлежать
Cooperates with	Взаимодействует с
Depicts	Описывает
Is assigned to	Связан (а) с
Is assigned 1:1 (1:n, n:m)	Имеет отношение 1:1 (1:n, n:m)
Is composed of	Состоит из
Is disciplinary superior to	Является непосредственным руководителем
Is located at	Располагается
Is managed by	Находится под управлением
Is of type	Относится к типу
Is organization manager for	Является организационным управляющим
Is position of	Является должностью
Is responsible for	Отвечает за
Is superior	Имеет в подчинении
Is technical superior to	Является техническим руководителем
Has assigned	Связан (а)с
Has member	Имеет в своем составе
Occupies	Занимает
Performs	Формирует
Substitutes for	Замещает
Subsumes	Содержит

### Моделирование бизнеса. Методология ARIS

Модель строится иерархически, от верхнего уровня структуры к нижнему. В модель верхнего уровня включаются самостоятельные подразделения, входящие в структуру организации. Каждое из них детализируется на более низкие уровни — уровни структурных подразделений.

Каждое структурное подразделение, в свою очередь, детализируется на структурные подразделения в его составе, которые изображаются на одной диаграмме.

Низшим уровнем является описание подразделений на уровне должностей — штатных единиц, занимаемых конкретными сотрудниками. При детализации моделей подразделений до уровня сотрудников целесообразно полностью указывать должность в составе подразделения. В случае, если в одном подразделении имеется несколько одинаковых должностей, то они нумеруются.

Пример организационной схемы приведен на рис. 94.

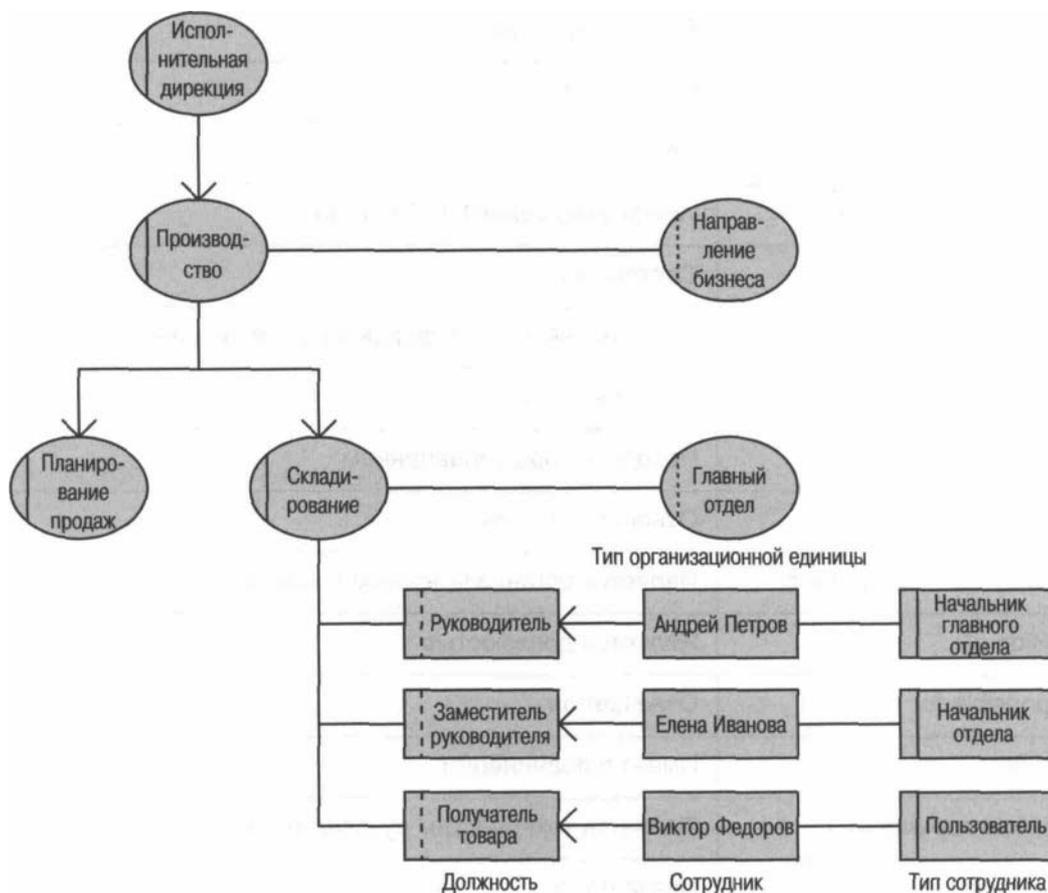


Рис. 94. Пример организационной схемы

## Простой методологический фильтр. Обзор моделей

Таблица 5. Атрибуты объектов организационной схемы

Атрибут	Наименование в системе ARIS	Описание атрибута
Имя	Name	Этот атрибут может изменяться непосредственно на графическом изображении объекта. Длина названия ограничена 40 символами. Задается обязательно
Полное имя	Full name	Длина полного названия ограничена 100 символами. Задается, только если имя содержит больше 40 символов
Идентификатор	Identifier	Этот параметр используется системой ARIS для идентификации объекта
Количество сотрудников	Number of employees	Количество сотрудников в подразделении или на должности
Описание/ Определение	Description/Definition	Используется для словесного описания краткого содержания объекта
Примечание/ Пример	Remark/Example	Количество заполняемых символов - не более 3000. Может быть приведен пример использования объекта или какие-либо примечания и комментарии. В случае, если данный атрибут заполняется разными пользователями, должен быть указан автор ремарки

Для объектов организационной схемы необходимо задать атрибуты, приведенные в табл.5.

Атрибуты Time of generation (Дата и время создания объекта). Creator (Автор),

Last change (Дата и время внесения последних изменений), Type (Тип модели). Last user (Последний пользователь) заполняются автоматически при создании объекта.

## 6.2. Функциональные модели

### 6.2.1. Дерево функций — Function Tree

*Функция* — описание элемента работы, образующего один логический этап в рамках процесса. В ARIS используется диаграмма «Дерево функций», посредством которой

функции могут быть описаны с различными уровнями детализации. При этом функции представляются не обязательно в хронологическом порядке.

На диаграммах функция обозначается прямоугольником зеленого цвета с скругленными краями (рис. 95). Возможные виды связей между функциями в модели «Дерево функций» приведены в табл.6.

**Таблица 6. Связи объектов диаграммы «Дерево функций»**

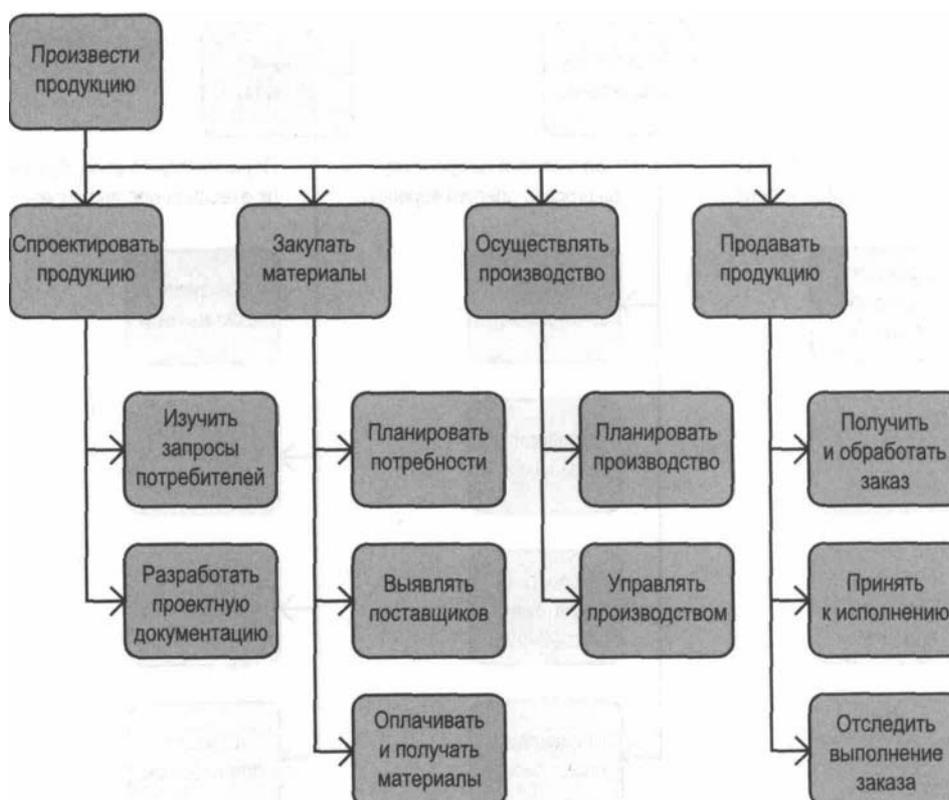
Английское название связи	Русское название связи
Is execution-oriented superior	Подчиняется по способу выполнения
Is object-oriented superior	Подчиняется по объекту
Is process-oriented superior	Подчиняется по процессу

На самом верхнем уровне описываются наиболее сложные функции, представляющие собой отдельный бизнес-процесс или процедуру. Детализация функций образует иерархическую структуру их описаний. Пример дерева функций приведен на рис. 95.

Для более содержательного позиционирования отдельного уровня иерархии в общей структуре функций наряду с понятием «функция» могут быть использованы также другие

термины: транзакция, процесс, подфункция, базовая функция (операция).

Разделение функций на элементы может происходить на нескольких иерархических уровнях. Базовые функции представляют самый нижний уровень в семантическом дереве функций. *Базовая функция* — это функция, которая уже не может быть разделена на составные элементы с целью анализа бизнес-процесса.



**Рис. 95. Дерево функций**

Функции объединяются в функциональное дерево в соответствии с различными критериями. Наиболее часто для этих целей используются такие критерии:

- объектно-ориентированный — обработка одного и того же объекта;

- процессно-ориентированный — принадлежность одному и тому же процессу;

- операционно-ориентированный — выполнение одинаковых операций.

Объектно-ориентированный подход при детализации функции иллюстрирует рис. 96а.

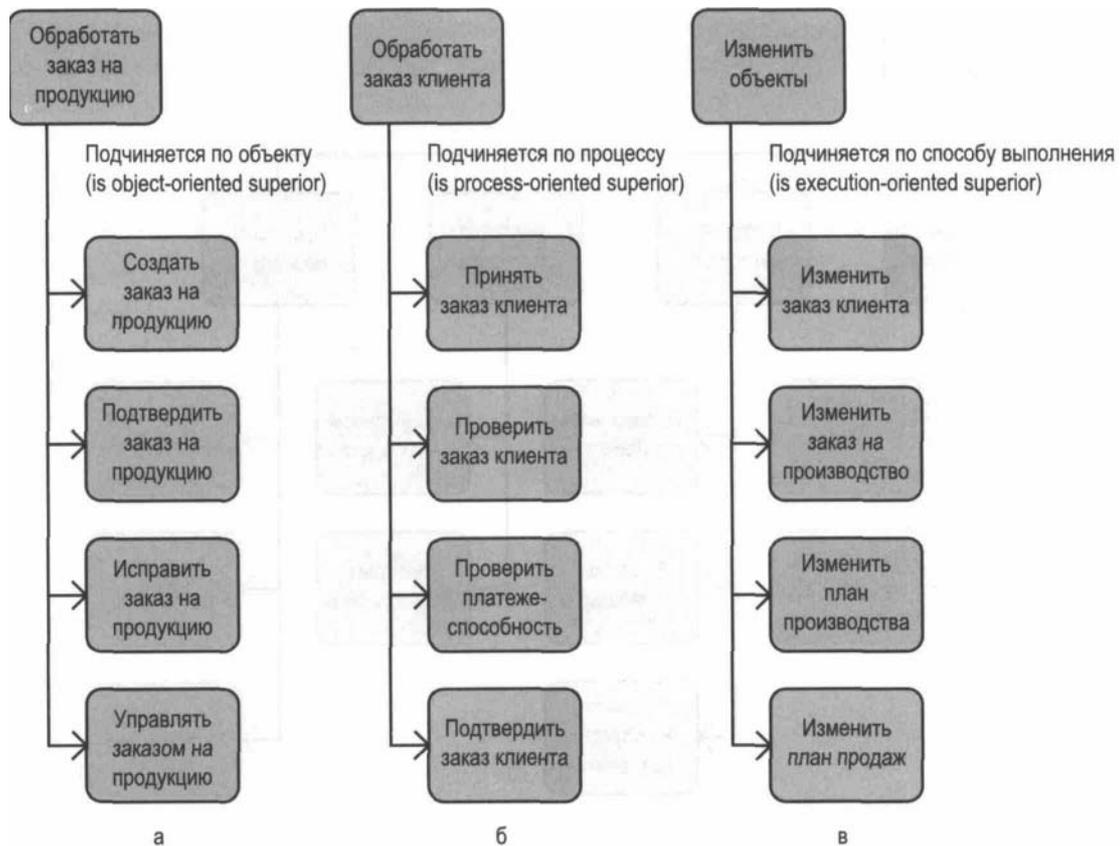


Рис. 96. Дерево функций: объектно-ориентированное (а); процессно-ориентированное (б); операционно-ориентированное (в)

Функция «Обработать заказ на продукцию» (рис. 96а) описывает различные операции (создать, подтвердить, исправить и т.д), которые выполняются над одним и тем же объектом, в данном случае — над объектом «Заказ на продукцию».

Если дерево функций используется в рамках моделирования бизнес-процесса, предпочтительнее применять метод, позволяющий построить процессно-ориентированное дерево. На рис. 96б представлена процессно-ориентированная детализация функции (последовательность функций, составляющих процесс). В отличие от объектно-ориентированного разбиения при процессно-ориентированной детализации

критерием служат операции, которые выполняются над различными объектами (заказ клиента, платежеспособность) в рамках одного бизнес-процесса.

При операционно-ориентированном подходе функция верхнего уровня декомпозируется на подфункции, каждая из которых выполняет ту же операцию, но с различными объектами. На рис. 96в приведена функция, выполняющая операцию «Изменить объекты». Функции могут принадлежать различным процессам и привлекаться к обработке различных объектов. Однако выполняемый ими тип операции над отдельными объектами всегда один и тот же.

### Простой методологический фильтр. Обзор моделей

Способ представления функций в виде дерева позволяет уменьшить степень сложности и является статичным описанием функции.

Свойства функции описываются ее многочисленными атрибутами, часть из которых показана на рис. 97.

Attribute	Value 1	Value 2
Name	Выпускать продукцию [Русский]	(Untitled) []
Identifier	Выпускать продукцию	
Full name		
Description/Definition		
Remark/Example		
Processing code		
Author		
Short description		
Type	Function	Function
Time of generation	20.12.00 16:44:17	20.12.00 16:44:17
Creator	system	system
Last change	20.12.00 16:44:32	20.12.00 16:44:32
Last user	system	system
Module code		
Hierarchy number		
Synonyms		

Рис, 97. Основные атрибуты функции

### Моделирование бизнеса. Методология ARIS

стандартах. Терминология в более узких областях знаний также должна быть систематизирована.

По этой причине набор методов ARIS содержит так называемую модель технических терминов, которая не только позволяет манипулировать различными терминами как синонимами, но и дает возможность поддерживать отношения между объектами в моделях данных.

Для представления этих отношений вводится тип объекта «технический термин». Теперь с каждым информационным объектом модели данных могут быть связаны разные технические термины (см. рис. 98).

Технические термины могут быть взаимосвязаны и иерархически упорядочены. Термины, определяемые рассматриваемой моделью, могут использоваться и в других диаграммах, которые содержат информационные объекты, например, в диаграммах процессов для представления входа/выхода данных функции.

## 6.3. Модели данных

### 6.3.1. Модель технических

#### терминов — Technical Term Models

При моделировании бизнеса часто приходится иметь дело с многочисленными терминами, определяющими информационные и иные объекты в организациях. Например, то, что понимается под термином «заказ» в отделе закупок, может значительно отличаться от того, что под этим подразумевают сотрудники производственного отдела. Введение соответствующей терминологии для организации и ее подразделений позволяет сделать информацию более понятной.

Термины и определения, используемые в отдельных науках (метрологии, электротехнике и других), узаконены в государственных

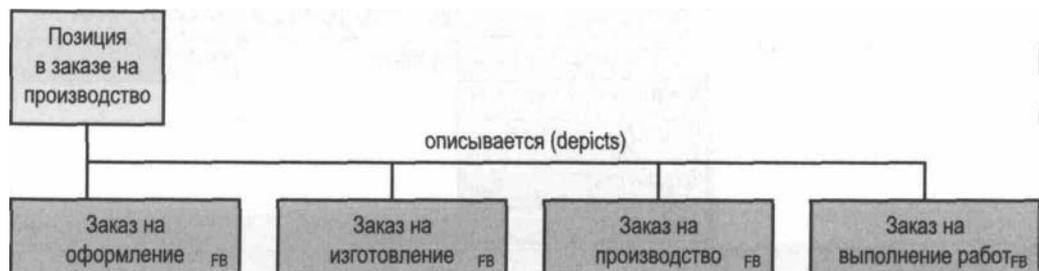


Рис. 98. Модель технических терминов

Объекты и их названия, используемые в модели технических терминов, приведены в табл. 7, а допустимые связи — в табл. 8.

На рис. 99 показана панель инструментов для создания модели в модуле *ARIS Designer*

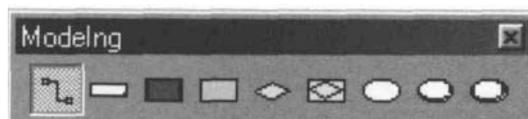


Рис. 99. Панель инструментов для создания моделей технических терминов

Таблица 7. Объекты модели технических терминов

	Английское название объекта	Русское название объекта
	Connection	Связь
	Technical term	Технический термин
	Cluster	Кластер
	Entity type	Тип сущности
	Relationship type	Тип отношения
	Reinterpretation relationship type	Переопределенный тип отношения
	D attribute (ERM)	Описательный атрибут (ERM)
	K attribute (ERM)	Ключевой атрибут (ERM)
	EK attribute (ERM)	Ключевой наследуемый атрибут (ERM)

Таблица 8. Связи объектов модели технических терминов

Английское название связи	Русское название связи
Can be	Может являться
Classifies	Классифицирует
Depicts	Отображает
Has	Имеет
Has relation with	Имеет отношение к
Is a	Является
Is a functional generic term of	Является практическим обобщающим термином для
Is feature of	Является свойством
Is generic term of	Является обобщающим термином для
Is part of	Является частью
Is specimen of	Является экземпляром
Subsumes	Содержит
Synonym (preferred term of)	Синоним (предпочтительный термин)

Все связи, кроме *depicts* (*отображает*), существуют только между объектами типа «технический термин». Связь *отображает* соединяет технические термины со всеми остальными объектами.

Тип связи *has relation with* (*имеет отношение к.*) отражает основные и однозначно классифицируемые отношения между двумя терминами предметной области.

Тип связи *is part of* (*является частью*) описывает двунаправленное отношение между двумя терминами предметной области. Эта связь указывает на то, что один из представленных терминов является составной частью другого; в противоположном направлении — термин имеет в своем составе другой термин.

Тип связи *is a* (*является*) устанавливает однозначное соответствие между двумя терминами предметной области.

Тип связи *classifies* (*классифицирует*) позволяет проводить группировку терминов. Группировка осуществляется за счет определения одного термина как подмножества экземпляров другого (родительского) термина. При этом родительский термин выступает в роли типа или класса.

Тип связи *is feature of* (*является свойством*) описывает двунаправленное отношение между двумя терминами предметной области. Он отражает тот факт, что один из терминов является отличительной характеристикой (свойством) другого термина. В противоположном направлении связь указывает на существование у заданного термина определенной отличительной характеристики.

Тип связи *can be* (*может являться*) означает, что один из указанных терминов может являться экземпляром из множества значений другого термина. Связь в противоположном направлении показывает, что экземпляр из множества значений одного термина может являться дополнительной характеристикой другого термина.

Тип связи *is specimen of* (*является экземпляром*) предназначен для отражения возможных экземпляров категории предлагаемых терминов.

Следует отметить, что любой тип связи в моделях ARIS является двунаправленным и, соответственно, имеет прямое и противоположное значения.

## 6.4. Модели процессов/управления

### 6.4.1. Событийная цепочка процесса — Extended event driven process chain (eEPC)

Событийная цепочка процесса (кратко — модель или диаграмма eEPC). Модель предназначена для детального описания процессов, выполняемых в рамках одного подразделения, несколькими подразделениями или конкретными сотрудниками. Она позволяет выявлять взаимосвязи между организационной и функциональной моделями. Модель eEPC отражает последовательность функциональных шагов (действий) в рамках одного бизнес-процесса, которые выполняются организационными единицами, а также ограничения по времени, налагаемые на отдельные функции.

Для каждой функции могут быть определены начальное и конечное события, ответственные исполнители, материальные и документарные потоки, сопровождающие модель, а также проведена декомпозиция на более низкие уровни (подфункции и т.д.). Модель eEPC является наиболее информативной и удобной при описании деятельности подразделений организации. Панель инструментов для создания модели eEPC показана на рис. 100. Объекты eEPC и их названия приведены в табл. 9. Связи между объектами диаграммы eEPC представлены в табл. 10.

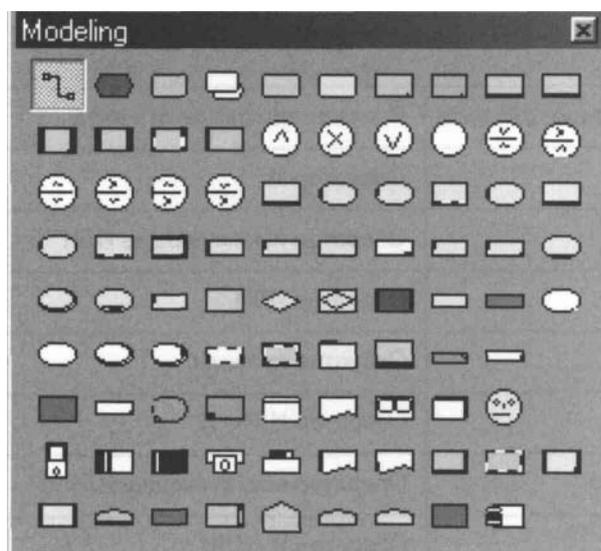


Рис. 100. Панель инструментов для создания моделей eEPC — событийных цепочек процессов

Таблица 9. Объекты модели eEPC

	Английское название объекта	Русское название объекта
	Connection	Связь
	Event	Событие
	Function	Функция
	Process interface	Интерфейс процесса
	Function actual	Функция существующая
	Function target	Функции планируемая
	System function, actual	Системная функция существующая
	System function, target	Системная функция планируемая
	IT function	Функция IT
	IT function type	Тип функции IT
	Application system	Прикладная система
	Application system class	Класс прикладной системы
	Application system type	Тип прикладной системы
	Hardware component type	Тип оборудования

Продолжение табл. 9

	Английское название объекта	Русское название объекта
	AND operator	Оператор И
	XOR operator	Оператор исключающее ИЛИ
	OR operator	Оператор ИЛИ
	Rule operator	Оператор правила
	OR/AND operator	Оператор ИЛИ/И
	XOR/AND operator	Оператор исключающее ИЛИ/И
	AND/OR operator	Оператор И/ИЛИ
	XOR/OR operator	Оператор исключающее ИЛИ/ИЛИ
	AND/XOR operator	Оператор И/исключающее ИЛИ
	OR/XOR operator	Оператор ИЛИ/исключающее ИЛИ
	Organizational unit	Организационная единица
	Organizational unit	Организационная единица
	Organizational unit type	Тип организационной единицы
	Organizational unit type	Тип организационной единицы
	System organization unit	Системная организационная единица
	System organization unit	Системная организационная единица
	System organization unit type	Тип системной организационной единицы
	System organization unit type	Тип системной организационной единицы
	Business object	Объект бизнеса
	Position	Должность
	Position type	Тип должности
	Internal person	Штатный сотрудник
	External person	Внештатный сотрудник
	Person type	Тип сотрудника
	Employee variable	Зарезервированная позиция

## Простой методологический фильтр. Обзор моделей

Продолжение табл. 9

	Английское название объекта	Русское название объекта
	Location	Местоположение
	Group	Группа
	Workstation	Рабочая станция
	Position description	Описание должности
	Entity type	Тип сущности
	Relationship type	Тип отношений
	Reinterpretation relationship type	Переопределенный тип отношений
	Cluster	Кластер
	Information actual	Информация фактическая
	Information target	Информация целевая
	Attribute type group	Группа типов атрибутов
	D attribute (ERM)	Описательный атрибут (ERM)
	K attribute (ERM)	Ключевой атрибут (ERM)
	EK attribute (ERM)	Ключевой наследуемый атрибут (ERM)
	COT attribute	COT Атрибут
	Complex object type	Тип сложного объекта
	Package	Пакет
	Class	Класс
	Operation	Операция
	Attributes	Атрибуты
	Object state	Состояние объекта
	Technical term	Технический термин
	Knowledge category	Категория знания
	Documented knowledge	Документированное знание
	File	Файл

Продолжение табл. 9

	Английское название объекта	Русское название объекта
	Document	Документ
	Magnetic tape	Магнитная лента
	Card file	Картотека
	Expertise	Экспертиза
	Folder	Папка
	Bar code	Штриховой код
	Microfiche	Микрофиша
	Telephone	Телефон
	Fax	Факс
	List	Список
	Draft list	Эскиз списка
	Screen	Экран
	Screen design	Тип экранного интерфейса
	Module	Модуль
	Module type	Тип модуля
	Product/Service	Продукт/услуга
	General resource	Общий ресурс
	Operating resource	Операционный ресурс
	Objective	Цель
	Service	Услуга
	Information service	Информационная услуга
	Authorization condition	Полномочия
	Component	Компонент

## Простой методологический фильтр. Обзор моделей

Таблица 10. Связи в модели eEPC

Английское название связи	Русское название связи
Accepts	Утверждает результат
Accesses	Имеет доступ к
Activates	Активизирует
Can access	Может иметь доступ к
Can be user	Может быть являться пользователем
Can support	Может поддерживать
Can use	Может использовать
Changes	Изменяет
Contributes to	Способствует при выполнении
Creates	Порождает
Creates output to	Создается на выходе
Decides on	Принимает решение по
Deletes	Уничтожает
Distributes	Распределяет
Executes	Выполняет
Has output of	Имеет на выходе
Has state	Отображает
Is approved during	Утверждается в ходе выполнения
Is archived by	Архивируется
Is checked by	Проверяется
Is consumed by	Потребляется
Is consolidation by	Оценивается с помощью
Is input for	Является входом для
Is IT responsible for	Отвечает по IT за
Is predecessor of	Предшествует

Продолжение табл. 10

Английское название связи	Русское название связи
Is responsible for development of	Отвечает за разработку
Is specimen owner of	Является владельцем экземпляра
Is technically responsible for	Отвечает за техническую часть
Is used by	Используется
Is user	Является пользователем
Leads to	Формирует
Must be information on cancellation	Должна быть информация о нестандартном завершении
Must be informed about	Должен быть информирован о выполнении
Must inform about result of	Должен информировать о результатах выполнения
Produces	Производит
Relates to	Имеет отношение
Supports	Поддерживает
Supports when time limit is exceeded	Поддерживает при превышении ограничения времени
Uses	Использует

Процедурная последовательность функций в рамках бизнес-процессов отображается в виде цепочки процесса, где для каждой функции могут быть определены начальное и конечное события. *Событие* — это состояние, которое является существенным для целей управления бизнесом и которое оказывает влияние или контролирует дальнейшее развитие одного или более бизнес-процессов.

События активизируют функции, т.е. передают управление от одной функции к другой. Они могут быть также результатом выполнения функций. В отличие от функций, которые имеют некоторую продолжительность, события происходят мгновенно.

Описание события должно содержать не только информационный объект («заказ»), но и описание изменения состояния («получен»). События переключают функции и могут быть результатом выполнения функции. Упорядочивание комбинации событий и функций в последовательность позволяет создать событийные цепочки процессов. С помощью этих диаграмм процедуры бизнес-процесса представляются как логические последовательности событий/функций.

Пример диаграммы событийной цепочки процесса приведен на рис. 101.

Можно рекомендовать следующие правила расположения графических элементов на диаграмме eEPC:

- графические элементы процесса (последовательность событий и функций) располагаются сверху вниз;
- графические элементы, обозначающие исполнителей функций (сотрудников или

Простой методологический фильтр. Обзор моделей

подразделения) располагаются справа от функций;

- документы, используемые при выполнении функций, а также формируемые в результате выполнения функций, располагаются слева от функций.

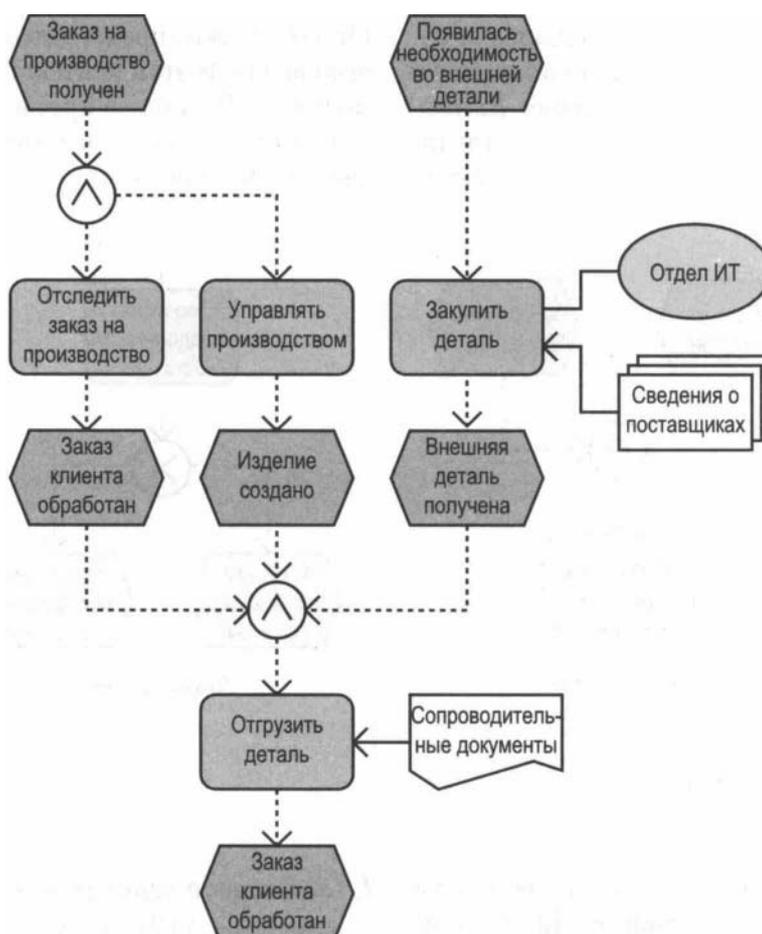


Рис. 101. Событийная цепочка процесса

### Моделирование бизнеса. Методология ARIS

Одно событие может инициировать выполнение одновременно нескольких функций, и наоборот, функция может быть результатом наступления нескольких событий. Эти ветвления и циклы обработки отображаются на диаграмме eEPC с помощью соединителей в виде небольшого кружка (см. рис. 102). Однако эти соединители не только отображают графические связи между элементами модели, но и определяют логические связи между объектами.

В первом случае на рис. 102 начальные события связываются с помощью оператора *AND* (*И*). Это означает, что функция *Выполнить операцию* запускается только тогда, когда маршрут доступен и проверено наличие

необходимых ресурсов. Другими словами, для начала выполнения функции должны произойти оба события.

Во втором случае показан оператор *XOR* (*исключающий ИЛИ*). Результатом выполнения функции *Проверить предложения поставщика* может быть принятие или отклонение его расценок. Однако оба события не могут произойти одновременно. Кроме указанных двух случаев и обычного оператора *OR* (*ИЛИ*), можно представить и более сложные отношения. В этом контексте можно ввести в модель eEPC общее правило, которое впоследствии будет описано более подробно в виде диаграммы правил.

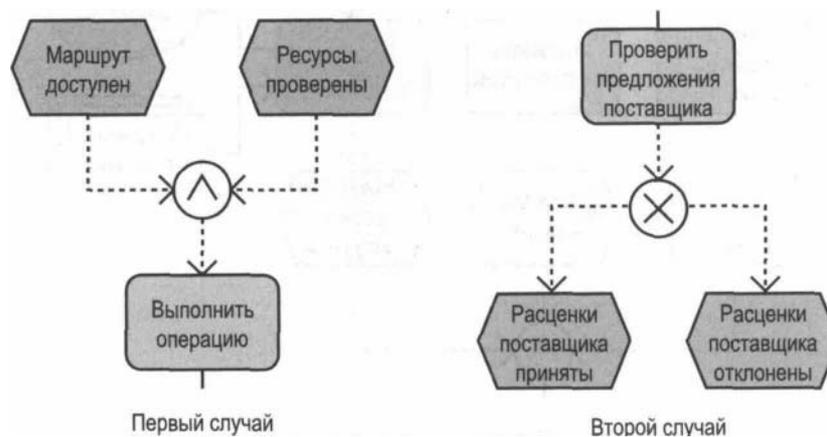


Рис.102. Примеры правил

Различают два типа операторов: операторы событий и операторы функций. На рис. 103. представлены все возможные операторы событий и функций.

Особое внимание необходимо уделить ограничениям, которые существуют для операторов функций. Поскольку события не могут принимать решения (в то время как функции могут), переключающееся событие не должно быть связано операторами *OR* или *XOR*!

Далее на примерах показано, какие операторы допустимы.

#### 1. Соединение переключающих событий

Оператор *AND* для переключающих событий (рис. 103а). Выполнение функции может быть начато после того, как произойдут все события.

Оператор *OR* для переключающих событий (рис. 103в). Эта функция выполняется, если произойдет по крайней мере одно событие.

Оператор *XOR* для переключающих событий (рис. 103д). Функция начинает выполняться после того, как произойдет одно (и только одно) событие.

## 2. Соединение сгенерированных событий

Оператор AND для сгенерированных событий (рис. 103б). В результате выполнения функции происходят все события.

Оператор OR для сгенерированных событий (рис. 103г). В результате выполнения

Простой методологический фильтр. Обзор моделей функции происходит по крайней мере одно событие.

Оператор XOR для сгенерированных событий (рис. 103е). В результате выполнения функции происходит максимум одно событие.

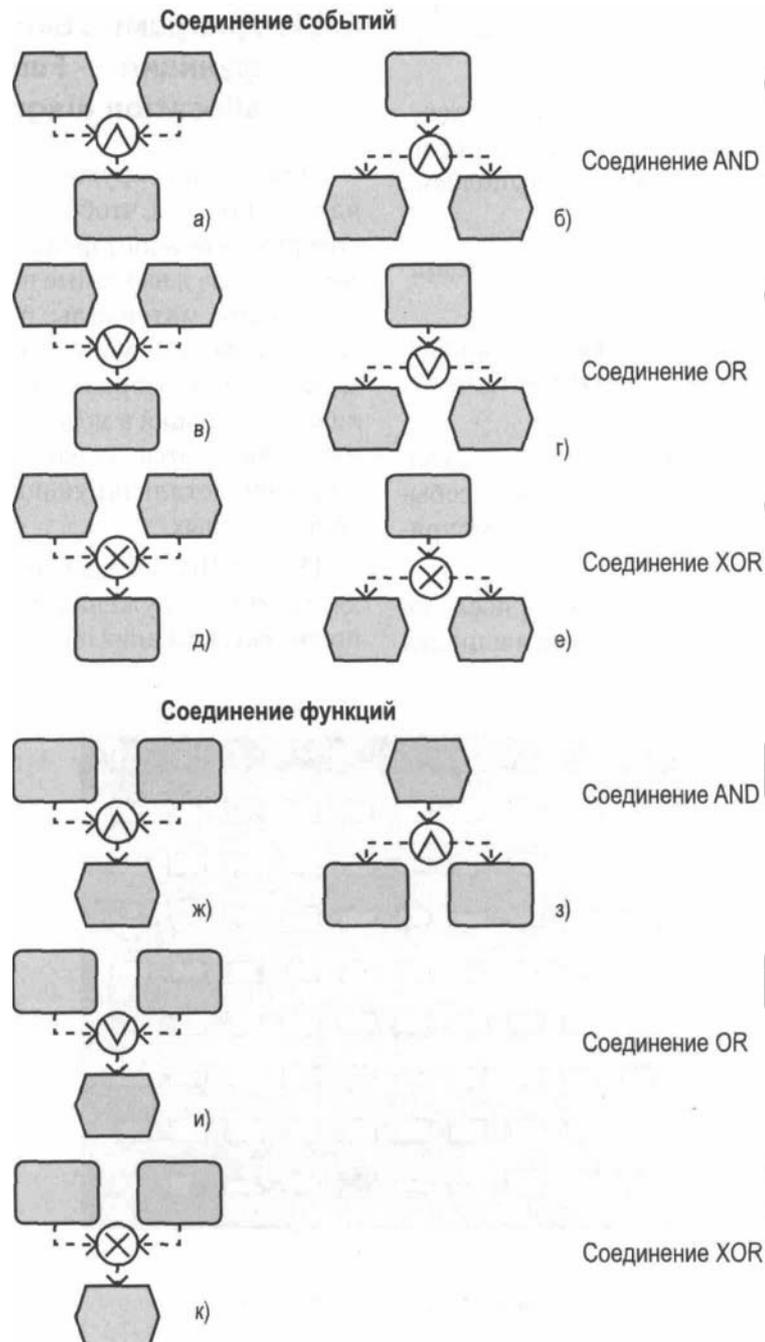


Рис. 103. Операторы соединений

3. Соединение функций со сгенерированными событиями.

Оператор AND для связи функций и сгенерированных событий (рис. 103ж). События происходят только после того, как все функции выполнены.

Оператор OR для связи функций и сгенерированных событий (рис. 103и). Событие произойдет после того, как будет выполнена по крайней мере одна функция.

Оператор XOR для связи функций и сгенерированных событий (рис. 103к). Событие произойдет после того, как будет выполнена одна (и только одна) функция.

4. Соединение функций с переключающими событиями

Оператор AND для связи функций и переключающих событий (рис. 103з). Событие переключает обе функции.

Операторы OR и XOR не могут использоваться для соединения переключающих событий и функций, так как события не могут принимать решения.

Поскольку функции вызываются последовательно, в модели eEPC ветвления и циклы обработки могут оказаться

представленными достаточно запутанным способом.

Модель eEPC имеет ряд разновидностей: в виде столбцов и строк, в виде таблиц, с потоком материалов.

#### 6.4.2. Диаграмма окружения функции — Function allocation diagram

Диаграмма окружения функции предназначена для того, чтобы описать все объекты, которые окружают функцию, — исполнителей, входные и выходные потоки информации, документы, материалы, продукты/услуги, а так же используемое оборудование. Этот тип моделей целесообразно применять для детализации функций в модели eEPC, в результате чего отражаются дополнительные связи и отношения, детализирующие эту функцию на уровне данных.

На рис. 104 показана панель инструментов диаграммы окружения функции, а в табл. 11 приведены названия объектов диаграммы.

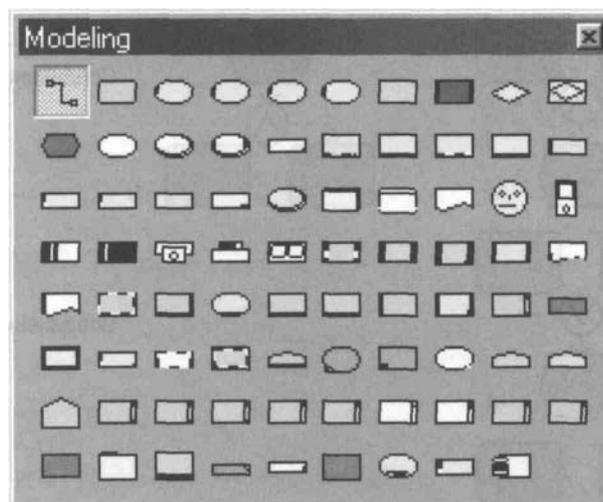


Рис. 104. Панель инструментов диаграммы окружения функции

Таблица 11. Объекты диаграммы окружения функции

	Английское название объекта	Русское название объекта
	Connection	Связь
	Function	Функция
	System organization unit type	Тип системной организационной единицы
	System organization unit	Системная организационная единица
	Organizational unit type	Тип организационной единицы
	Organizational unit	Организационная единица
	Entity type	Тип сущности
	Cluster	Кластер
	Relationship type	Тип отношений
	Reinterpretation relationship type	Переопределенный тип отношений
	Event	Событие
	D attribute (ERM)	Описательный атрибут (ERM)
	K attribute (ERM)	Ключевой атрибут (ERM)
	EK attribute (ERM)	Ключевой наследуемый атрибут (ERM)
	Technical term	Технический термин
	System organization unit type	Тип системной организационной единицы
	System organization unit	Системная организационная единица
	Organizational unit type	Тип организационной единицы
	Organizational unit	Организационная единица
	Position	Должность
	Position type	Тип должности
	Person type	Тип сотрудника
	Internal person	Штатный сотрудник
	External person	Внештатный сотрудник
	Group	Группа

Продолжение табл. 11

	Английское название объекта	Русское название объекта
	Card file	Картотека
	File	Файл
	Document	Документ
	Expertise	Экспертиза
	Folder	Папка
	Bar code	Штриховой код
	Microfishe	Микрофиша
	Telephone	Телефон
	Fax	Факс
	Magnetic tape	Магнитная лента
	Application system class	Класс прикладной системы
	Application system type	Тип прикладной системы
	Application system	Прикладная система
	Module type	Тип модуля
	Draft list	Эскиз списка
	List	Список
	Screen design	Тип экранного интерфейса
	Screen	Экран
	Location	Местоположение
	IT function type	Тип IT функции
	IT function	Функция IT
	Hardware component type	Тип оборудования
	Module	Модуль
	Operating resource	Операционный ресурс
	General resource	Общий ресурс

## Простой методологический фильтр. Обзор моделей

Продолжение табл. 11

	Английское название объекта	Русское название объекта
	Business object	Бизнес-объект
	Employee variable	Зарезервированная позиция
	COT attribute	COT атрибут
	Complex object type	Сложный тип объекта
	Product/Service	Продукт/услуга
	Knowledge category	Категория знания
	Documented knowledge	Документированное знание
	Attribute type group	Группа типов атрибутов
	Service	Услуга
	Information service	Информационная услуга
	Objective	Цель
	Warehouse equipment	Складское оборудование
	Technical operating supply	Вспомогательные технические средства
	Transport systems	Транспортная система
	Operating resource type	Тип операционного ресурса
	Warehouse equipment type	Тип складского оборудования
	Material type	Тип материала
	Packaging material type	Тип упаковочного материала
	Technical operating supply type	Тип вспомогательных технических средств
	Transport system type	Тип транспортной системы
	Authorization condition	Полномочия
	Package	Пакет
	Class	Класс
	Operation	Операция
	Attributes	Атрибуты

Продолжение табл. 11

	Английское название объекта	Русское название объекта
■	Object state	Состояние объекта
□	Workstation	Рабочая станция
▢	Position description	Описание должности
▣	Component	Компонент

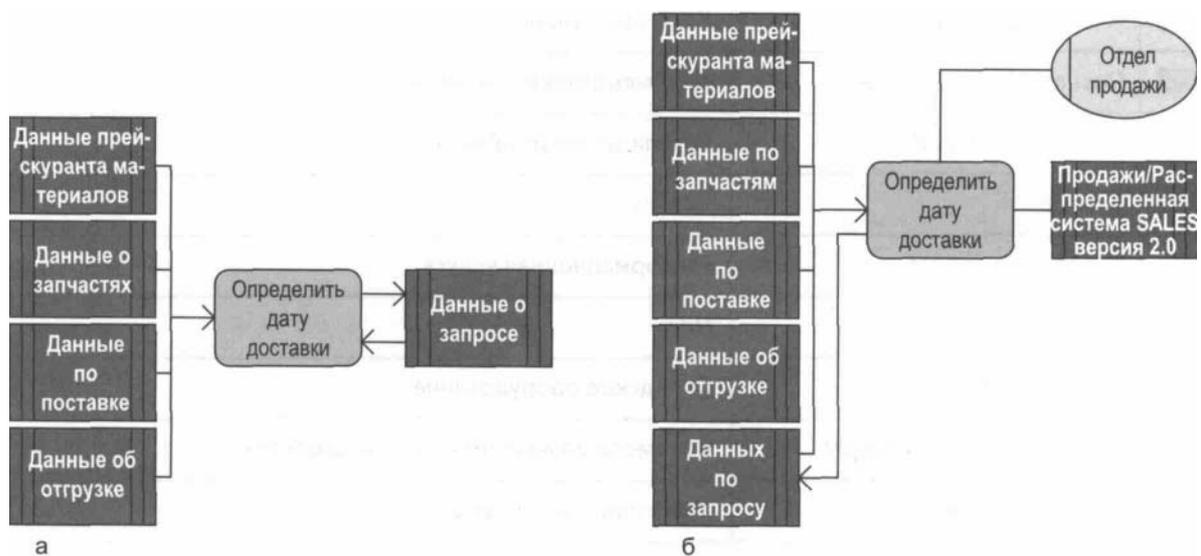


Рис. 105. Пример диаграммы окружения функции

Виды связей в диаграммах окружения функции практически совпадают со связями диаграммы eEPC.

Диаграммы окружения функции используются для того, чтобы уменьшать сложность диаграмм eEPC. Они дают возможность в деталях описывать статические отношения между функциями и объектами других моделей. На рис. 105 показан пример диаграммы окружения функции. Данная диаграмма описывает объекты, необходимые для успешного выполнения функции, а также входные и выходные данные.

Диаграмма окружения функции представляет собой возможный тип связи между моделью данных и функциональной моделью. В ней, как правило, отражаются преобразование входных данных в выходные и представление потока данных между отдельными функциями.

Диаграммы окружения функций содержат функции из функциональной модели и информационные объекты из модели данных.

Можно провести более детальную спецификацию, указывающую, создает или удаляет отдельная функция информационный объект. В зависимости от степени детализации информационные объекты могут представлять кластеры данных, сущности или типы отношений, а также атрибуты модели данных.

Кроме входных/выходных данных, функций и событий доступны и могут быть исполь-

зованы все другие объекты, связанные с отдельной функцией в диаграмме eEPC. Таким образом, пользователь при моделировании процесса с помощью диаграмм eEPC может ограничиться описанием событий и функций, затем связать с функцией диаграмму окружения функции и уже в ней отразить дополнительные детализирующие связи и отношения. Это позволяет отображать бизнес-процессы более четко, а также поясняет названия нового типа модели.

### 6.4.3. Производственный и офисный процессы — Industrial and Office process

Модели производственного и офисного процессов описывают практически те же процессы, что и модель eEPC. Однако эти две модели используют ограниченное число

### Простой методологический фильтр. Обзор моделей

объектов и символов, которые представляются наглядными, легко понимаемыми пиктограммами.

Преимущество пиктографического представления состоит в том, что сотрудники различных подразделений могут не только понимать эти модели без предварительной подготовки, но и самостоятельно их разрабатывать и при необходимости изменять. Диаграммы этих двух типов позволяют проводить моделирование, анализ и оптимизацию процесса непосредственно в подразделениях. Производственный процесс представляет процессы создания материальных продуктов/услуг, а офисный процесс отображает процессы, протекающие в офисе (создание нематериальных продуктов/ту слуг).

В табл. 12 сравниваются символы моделей производственного и офисного процессов с символикой моделей eEPC.

Таблица 12. Изображение объектов в моделях производственного и офисного процессов и в модели eEPC

Тип объекта	Символы моделей		
	eEPC	производственный процесс	офисный процесс
Событие			
Функция		 Функция (Производственный процесс)	 Функция (Офисный процесс)
Правило	 AND XOR OR Правило		
Тип прикладной системы			
Местоположение	 Местоположение		

Продолжение табл. 12

Тип объекта	Символы возможных типов диаграмм		
	еЕPC	производственный процесс	офисный процесс
Организационная единица			
Группа			
Должность			
Тип сотрудника	 Тип сотрудника		
Сотрудник	 Штатный сотрудник	 Сотрудник (м)    Сотрудник (ж)	 Сотрудник (м)    Сотрудник (ж)
Категория знаний			
Документированные знания			
Носитель информации	 Штрих-код    Файл Факс    Карточка Магнитная лента    Микрофиша Телефон    Документ Папка    Экспертная оценка	 Принтер    Документ    Дискета Шкаф    Блокнот    Телефон Полки    Электронная почта	 Принтер    Документ    Дискета Шкаф    Блокнот    Телефон Накопитель файлов    Эл. почта    Планировать время Корзина для мусора    Интернет    Книга Жесткий диск    CD-ROM    Факс Письмо    Папка    Магнитная лента

### Простой методологический фильтр. Обзор моделей

Можно переходить от одного типа представления к другому в рамках моделей этих трех типов. В процессе копирования ARIS автоматически конвертирует символы в

соответствии с типом диаграммы. На рис. 106 представлено описание одних и тех же фактов моделями трех типов — eEPC, производственного и офисного процессов.

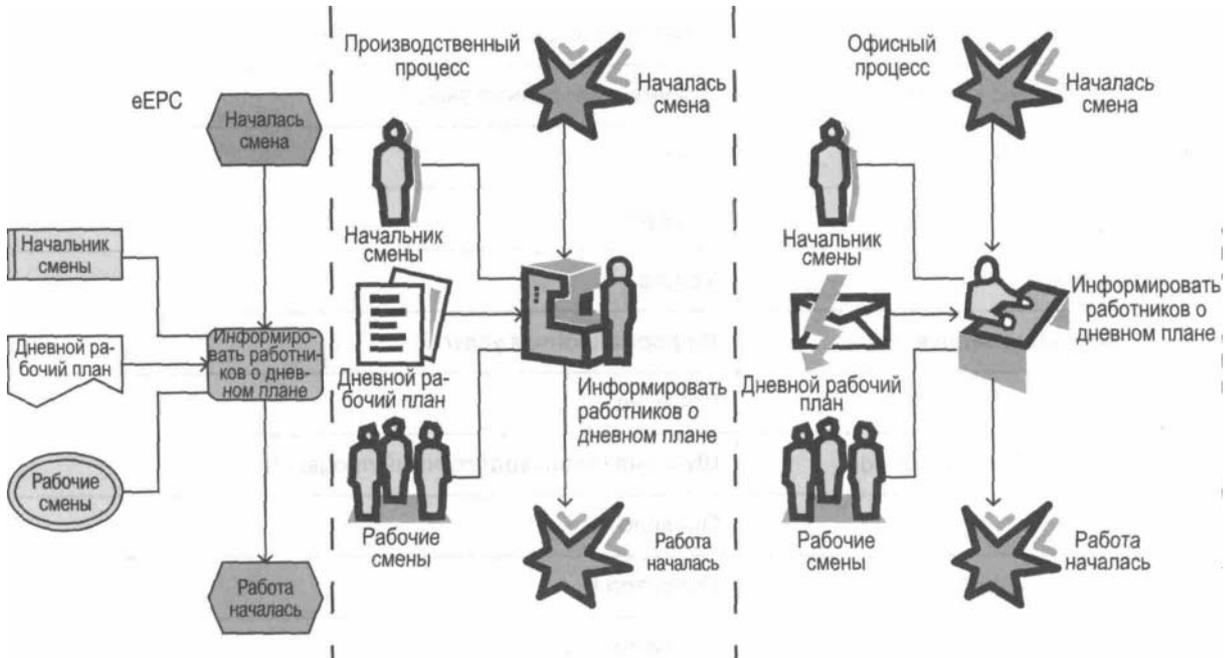


Рис. 106. Представление одних и тех же фактов с помощью моделей eEPC, производственного и офисного процессов

На рис. 107 показана панель инструментов, используемая при моделировании производственного процесса. Изображения и названия объектов приведены в табл. 13. Изображения и названия объектов для

моделирования офисного процесса приведены в табл. 14.

На рис. 108 показана панель инструментов, используемая при создании моделей офисных процессов.



Рис. 107. Панель инструментов для моделирования производственного процесса

Таблица 13. Объекты производственного процесса

	Английское название объекта	Русское название объекта
	Connection	Связь
	Knowledge category	Категория знания
	Documented knowledge	Документированное знание
	Objective	Цель
	Product	Продукт
	Service	Услуга
	Information service	Информационная услуга
	Event	Событие
	Function (Manufacturing)	Функция (производственный процесс)
	Rule	Правило
	AND operator	Оператор И
	OR operator	Оператор ИЛИ
	XOR operator	Оператор исключаящее ИЛИ
	Printer	Принтер
	Document	Документ
	Diskette	Дискета
	File cabinet	Картотечный шкаф
	Notepad	Блокнот
	Telephone	Телефон
	E-mail	Электронная почта
	File bin	Накопитель файлов
	Location	Местоположение
	Position	Должность
	Person (m)	Сотрудник (мужчина)
	Person (f)	Сотрудник (женщина)

## Простой методологический фильтр. Обзор моделей

Продолжение табл. 13

	Английское название объекта	Русское название объекта
	Person type	Тип сотрудника
	Organizational unit	Организационная единица
	Group	Группа
	Packaging material	Упаковочный материал
	Airplane	Самолет
	Truck	Грузовик
	Transport System	Транспортная система
	Machine	Станок
	Robot	Робот
	Warehouse Equipment	Складское оборудование
	Tool	Инструменты
	Material Type	Тип материала
	Material Type	Тип материала
	Computer	Компьютер
	General resource	Общий ресурс
	Entity type	Тип сущности
	Relationship type	Тип отношений
	ERM attribute	ERM атрибут
	Class	Класс
	Cluster	Кластер
	Technical term	Технический термин
	Product/Service	Продукт/услуга
	Screen	Экран
	Package	Пакет



Рис. 108. Панель инструментов для моделирования офисного процесса

Таблица 14. Объекты офисного процесса

	Английское название объекта	Русское название объекта
	Connection	Связь
	Knowledge category	Категория знания
	Documented knowledge	Документированное знание
	Objective	Цель
	Product	Продукт
	Service	Услуга
	Information service	Информационная услуга
	Event	Событие
	Function (Office)	Функция (офисный процесс)
	Rule	Правило
	AND operator	Оператор И
	OR operator	Оператор ИЛИ
	XOR operator	Оператор исключающее ИЛИ
	CD-ROM	Оптический диск
	Hard disk	Жесткий диск
	File	Файл
	Document	Документ

Продолжение табл. 14

	Английское название объекта	Русское название объекта
	Wastepaper basket	Корзина для ненужных бумаг
	Letter	Письмо
	Diskette	Дискета
	Book	Книга
	File cabinet	Картотечный шкаф
	Notepad	Блокнот
	Magnetic tape	Магнитная лента
	Internet	Интернет
	Fax	Факс
	Telephone	Телефон
	E-mail	Электронная почта
	File bin	Накопитель файлов
	Printer	Принтер
	Time planner	Ежедневник
	Location	Местоположение
	Position	Должность
	Person (m)	Мужчина
	Person (f)	Женщина
	Person type	Тип служащего
	Organizational unit	Организационная единица
	Group	Группа
	Computer	Компьютер
	General resource	Общий ресурс
	Entity type	Тип сущности
	Relationship type	Тип отношений

Продолжение табл. 14

	Английское название объекта	Русское название объекта
	ERM attribute	ERM атрибут
	Class	Класс
	Cluster	Кластер
	Technical term	Технический термин
	Product/Service	Продукт/услуга
	Screen	Экран
	Package	Пакет

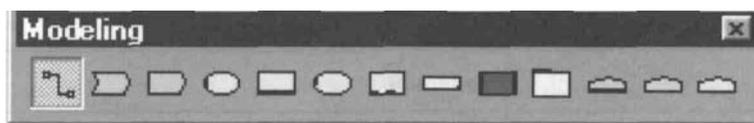


Рис. 109. Панель инструментов диаграммы цепочек добавленного качества

Таблица 15. Объекты диаграммы цепочек добавленного качества

	Английское название объекта	Русское название объекта
	Connection	Связь
	Value-added chain	Звено цепочки добавленного качества
	Value-added chain	Звено цепочки добавленного качества
	Organizational unit	Организационная единица
	Organizational unit	Организационная единица
	Organizational unit type	Тип организационной единицы
	Organizational unit type	Тип организационной единицы
	Technical term	Технический термин
	Cluster	Кластер
	Package	Пакет

Продолжение табл. 15

	Английское название объекта	Русское название объекта
	Product/Service	Продукт/услуга
	Service	Услуга
	Information service	Информационная услуга

Виды связей, используемых в моделях производственного и офисного процессов, практически совпадают со связями, используемыми отображаемыми в модели ePC.

#### 6.4.4. Диаграмма цепочек добавленного качества — Value-added chain diagram (VAD)

Диаграмма цепочек добавленного качества описывает функции организации, которые непосредственно влияют на реальный выход ее продукции. Эти функции создают

последовательность действий, формируя добавленные значения: стоимость, количество, качество и т.д.

Аналогично дереву функций описываемые функции могут размещаться в диаграмме согласно иерархическому принципу, т.е. наиболее важные функции располагаются левее и выше. Эта иерархия всегда иллюстрирует подчинение функций. Кроме этого, рассматриваемая диаграмма может представлять связи между функциями, организационными единицами и преследуемыми целями.

**Таблица 16. Связи объектов диаграммы цепочек добавленного качества**

Английское название связи	Русское название связи
Accepts	Утверждает результат
Contributes to	Способствует при выполнении
Executes	Выполняет
Is consumed by	Потребляется
Decides on	Принимает решение по
Has consulting role in	Участвует в качестве консультанта
Has output of	Имеет на выходе
Is input for	Является входом для
Is IT responsible for	Отвечает по IT за
Is process-oriented superior	Подчиняется по процессу

Продолжение табл. 16

Английское название связи	Русское название связи
Is predecessor of	Предшествует
Is superior	Имеет в подчинении
Is technical responsible for	Отвечает за техническую часть
Must be informed on cancellation	Должен быть информирован о нестандартном завершении
Must be informed about	Должен быть информирован о выполнении
Must inform about result of	Должен информировать о результатах выполнения
Produces	Производит
Supports	Поддерживает

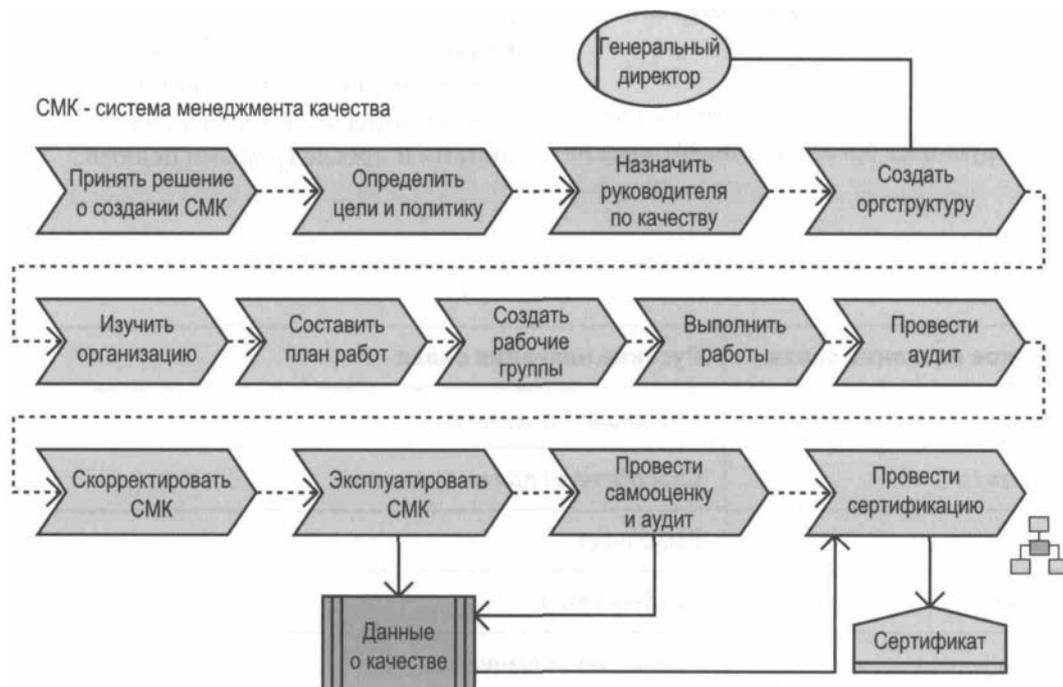


Рис. 110. Диаграмма цепочек добавленного качества для процесса построения системы менеджмента качества

Панель инструментов этой модели показана на рис. 109, названия объектов приведены в табл. 15, а допустимые виды связи — в табл. 16.

Следует отметить, что кластер используется для представления сложных объектов и является совокупностью некоторого количества связанных типов сущностей. Кластеры применяются для указания структуры информации, изменение которой фиксируется в событии.

Пример диаграммы цепочки добавленного качества приведен на рис. 110.

Построение диаграммы цепочки добавленного качества целесообразно начать с обзорного представления взаимосвязанных частей процесса путем расположения элементов процесса согласно временной последовательности их выполнения.

На следующем этапе рекомендуется отразить взаимозависимость различных элементов процесса путем нанесения соответствующих связей. После отображения в модели структуры процесса каждый из элементов процесса рассматривается с точки зрения необходимости его детализации. Если это требуется, то элемент детализируется на соответствующие блоки.

На завершающем этапе в модель процесса добавляются элементы организационной структуры, отвечающие за выполнение процесса или участвующие в процессе, а также документы, используемые в процессе.

К наиболее важным атрибутам модели следует отнести:

- *Name (Имя модели)*. Этот атрибут должен кратко и четко отражать назначение модели и быть уникальным. Количество заполняемых символов — не более 80.

#### Простой методологический фильтр. Обзор моделей

Задание этого атрибута обязательно. По умолчанию поле атрибута *Имя* соответствует названию модели, определенному пользователем при ее создании.

- *Description/Definition (Описание/Определение)*. Этот атрибут используется для описания краткого содержания модели или отображаемого в ней процесса. В данном поле целесообразно отразить специфику процесса. Если в полях атрибутов *Имя* и *Полное имя* недостаточно символов для отражения полного названия модели, то его можно указать в поле «Описание» модели. Количество заполняемых символов — не более 3000.

- *Fullname (Полное имя)*. Заполняется в случае, если атрибут *Имя* из-за длины представлен с сокращениями. Количество заполняемых символов — не более 100. Заполнение обязательно. Если для введения полного имени модели символов недостаточно, то используется атрибут *Description/Definition*.

- *Remark/Example (Примечание/Пример)*. Может быть приведен пример использования модели или какие-либо примечания и комментарии. Если данный атрибут заполняется разными пользователями, то должен быть указан автор ремарки. Количество заполняемых символов — не более 3000.

Атрибуты *Time of generation* (Дата и время создания модели), *Creator* (Автор), *Last change* (Дата и время последних изменений), *Type* (Тип модели) и *Last user* (Последний пользователь) заполняются автоматически при создании модели.

## 7. Стандартный и расширенный методологические фильтры. Обзор моделей

В данный фильтр, помимо рассмотренных далее моделей, входит организационная схема (см. раздел 6.1.1).

### 7.1. Функциональные модели

Дерево функций, входящее в состав данного фильтра, рассмотрено ранее (см. раздел 6.2.1).

#### 7.1.1. Диаграмма целей — Objective diagram

Диаграмма целей используется для описания целей организации и построения их иерархии. Объекты этой диаграммы представлены в табл. 17, а, возможные связи между ними — в таб. 18.

Панель инструментов показана на рис. 111.



Рис. 111. Панель инструментов для создания диаграммы целей

Таблица 17. Объекты диаграммы целей

	Английское название объекта	Русское название объекта
	Connection	Связь
	Objective	Цель
	Function	Функция
	Critical factor	Критический фактор
	Product/Service	Продукт/услуга
	Service	Услуга
	Information service	Информационная услуга

## Стандартный и расширенный методологические фильтры. Обзор моделей

Таблица 18. Связи объектов в диаграмме целей

Английское название связи	Русское название связи
Belongs to	Принадлежит
Has output of	Имеет на выходе
Is consumed by	Потребляется
Is input for	Является входом для
Is superior	Имеет в подчинении
Is used by	Используется
Produces	Производит
Supports	Поддерживает

Прежде чем начинать моделирование деятельности организации, нужно определить и описать ее цели, а также понять, насколько необходима реорганизация и оптимизация бизнеса.

Для этого используют диаграмму целей, в которой описывают цели организации и создают их иерархии.

*Цель* — желаемое будущее состояние организации, которое может быть достигнуто при выполнении выделенного состава функций с учетом критических факторов успеха.

В диаграмме целей необходимо определить критические факторы успеха, упорядочить их в иерархии и распределить в соответствии целями, которые они поддерживают. *Критические факторы успеха* — это движущие силы бизнес-процессов. Они позволяют оценить риски и сложности при достижении цели организации. Пример диаграммы целей приведен на рис. 112.

Этот тип диаграммы связан с другими посредством объекта «Функция». Каждой цели можно поставить в соответствие функцию, которая ведет к достижению данной цели.



Рис. 112. Диаграмма целей

### Моделирование бизнеса. Методология ARIS

Объекты диаграммы типа прикладной системы приведены в табл. 19, а возможные связи между ними — в табл. 20. На рис. 113 показана панель инструментов, используемая при построении такой диаграммы.

#### 7.1.2. Диаграмма типа прикладной системы — Application system type diagram (ASTD)

Данная диаграмма предназначена для моделирования прикладных информационных систем, используемых в организации.

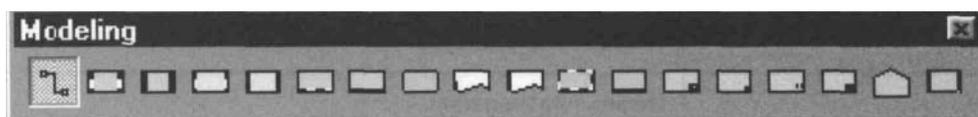


Рис. 113. Панель инструментов для создания диаграммы типа прикладной системы

Таблица 19. Объекты диаграммы типа прикладной системы

	Английское название объекта	Русское название объекта
	Connection	Связь
	Application system class	Класс прикладной системы
	Application system type	Тип прикладной системы
	Module class	Класс модуля
	Module type	Тип модуля
	IT function class	Класс функции IT
	IT function type	Тип функции IT
	Function	Функция
	Draft list	Эскиз списка
	List	Список
	Screen design	Тип экранного интерфейса
	Screen	Экран
	GUI (Graphical user interface) type	Тип графического интерфейса пользователя
	Operating system type	Тип операционной системы
	DBMS type	Тип СУБД
	Programming language	Язык программирования

## Стандартный и расширенный методологические фильтры. Обзор моделей

Продолжение табл. 19

	Английское название объекта	Русское название объекта
	Objective	Цель
	Program module type	Тип программного модуля

Таблица 20. Связи объектов в диаграмме типа прикладной системы

Английское название связи	Русское название связи
Belongs to	Принадлежит
Belongs to class	Принадлежит к классу
Can be platform for	Может являться платформой для
Can be supported by	Может поддерживаться
Can run under	Может быть запущен на
Can support	Может поддерживать
Creates	Порождает
Is created by	Порождается
Is predecessor of	Предшествует
Is represented by	Представляется
Is successor of	Следует за
Is supported by using	Поддерживается при использовании
Represents	Отображает
Subsumes	Содержит
Supports	Поддерживает

Иерархическая структура прикладной информационной системы представляет модульную структуру типовой информационной сис

темы, а также отражает реализацию отдельных функций бизнес-процессов. На рис. 34 показана структура инструментальной среды ARIS в виде диаграммы типа прикладной системы.

Основными понятиями, связанными с информационными технологиями, являются:

- *тип прикладной информационной системы* — обобщение совокупности систем, имеющих одинаковое предназначение и схожие технические характеристики;
- *тип модуля* — обобщение отдельных модулей, которые представляют собой независимо исполняемые части прикладной системы;
- *тип информационной функции (транзакции)* — обобщение отдельных транзакций, обладающих схожими техническими характеристиками;
- *класс прикладных информационных систем* — совокупность типов систем, отобранных по некоторому признаку классификации;
- *класс модулей* — совокупность типов модулей, соответствующих выбранному критерию;
- *класс информационных функций* — совокупность типов информационных функций;
- входы и выходы информационных функций;
- дизайн экранов.

Данный тип модели позволяет при моделировании ответить на следующие вопросы:

- Для достижения каких целей создается информационная система?
- В каких информационных системах, модулях и функциях реализуются функции бизнес-процессов, выделенные на уровне описания требований?
- Какая модульная структура существует у анализируемых систем?
- Какая выводимая информация и какой дизайн экрана необходимы пользователю?
- Какая информация создается в системе и как она используется?
- Какие технические особенности присущи информационной системе (операционная система, интерфейсы, используемая СУБД)?

- Какова может быть поддержка функций, определенных с помощью типов прикладных систем, типов модулей или проектов этих функций?

- Какие списки и экраны потребуются для выполнения функции?

- Какие списки могут быть созданы с помощью прикладной системы данного типа или модуля данного типа и какие экраны поддерживают прикладную систему и модули данных типов?

- Какая технологическая база имеется в распоряжении для реализации прикладной системы данного типа (операционная система, интерфейс пользователя или СУБД)?

Тип прикладной системы может рассматриваться как тип основного объекта в рамках функциональной модели на этапе спецификации проекта. Следует отметить, что конкретная прикладная система может рассматриваться только на уровне описания реализации функциональной модели. Например, прикладная система компании будет идентифицирована номером лицензии. В то же время тип прикладной системы формируется посредством объединения всех прикладных систем, базирующихся на одной и той же технологии. Тип прикладной системы — это некоторая обобщенная прикладная система, состоящая, как правило, из отдельных модулей. Диаграмма типа прикладной системы — это представление ее модульной структуры. В ARIS типы модулей образуют отдельные части типа системы, как показано на рис. 114.

Инструментальная среда ARIS относится к классу информационных систем для моделирования бизнеса и может работать в операционной системе Window 9x.

Сама система может содержать следующие модули: *ARIS Toolset* и *ARIS Web Publisher*. Так же, как типы прикладной системы, типы модулей объединяют отдельные модули, базирующиеся на одной и той же технологии. Типы модулей содержатся в типах прикладной системы и представляют отдельные выполняемые компоненты.

## Стандартный и расширенный методологические фильтры. Обзор моделей

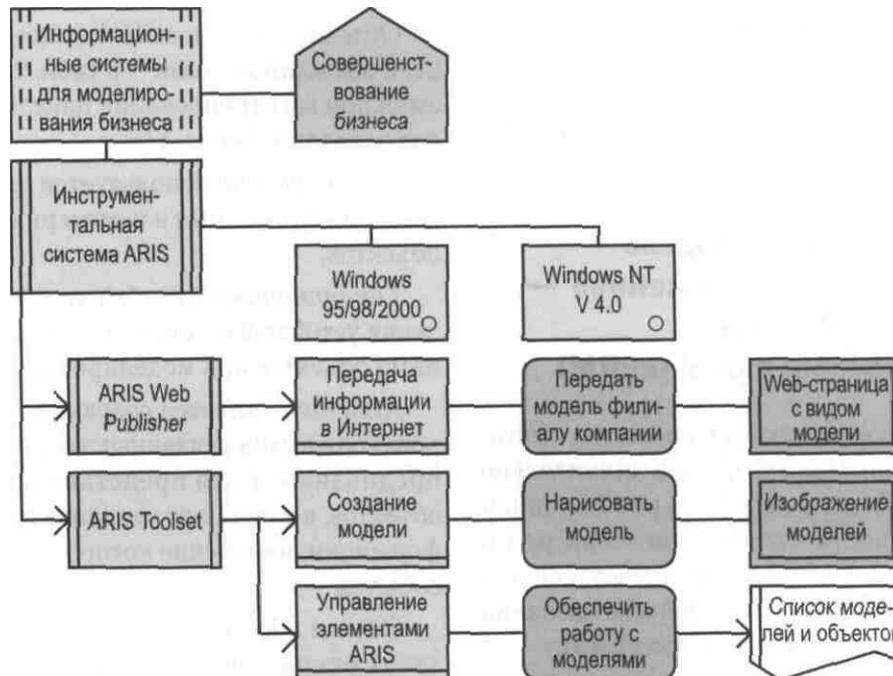


Рис. 114. Диаграмма типа прикладной информационной системы

Типы прикладных систем и типы модулей могут быть иерархически упорядочены любым способом. На самом нижнем уровне типы модулей можно декомпозировать на типы функций.

Типы функций — наименьшие элементы модуля, имеющие смысл транзакции. Они реализуются с помощью отдельных программных элементов и должны быть полностью выполняемыми, причем так, чтобы их выполнение происходило за единственный шаг.

С помощью диаграммы типа прикладной системы отдельные функции могут быть соотнесены и поддержаны определенными типами систем и модулей. Это позволяет отразить связь между сформулированными требованиями и спецификацией проекта в рамках функциональной модели.

Для того чтобы дальше специфицировать технологическую базу для типов прикладной системы и типов модулей, им могут быть приспаны соответствующие типы пользовательских интерфейсов, СУБД и операционных систем, а также языков программирования, которые использовались для их реализации.

Таким образом, тип прикладной системы может соответствовать пользовательскому интерфейсу Windows 98 и Windows NT (рис. 114). Это означает, что данная версия типа прикладной системы способна работать с пользовательскими интерфейсами обоих типов. Только в том случае, когда пользовательский интерфейс соответствует конкретной прикладной системе на уровне описания реализации в рамках функциональной модели, становится необходимой уникальная взаимосвязь. Эта взаимосвязь описывает конфигурацию отдельной лицензии на типы систем, которые были приобретены компанией.

Выполнение объектно-ориентированной функции с помощью прикладной системы подразумевает возможность использования различных экранов, создание списков и работу с ними. Поэтому можно ввести такие объекты, как *Список* и *Экран*, которые связываются либо с объектно-ориентированной функцией, либо с типами прикладной системы и типами модулей.

## 7.2. Модель данных

Модель технических терминов рассмотрена ранее в составе моделей простого методологического фильтра (см. раздел 6.3.1).

### 7.2.1. Расширенная модель «сущность - отношение» — Extended entity - relationship model (eERM)

В течение последних нескольких лет оригинальная модель Чена [15] была значительно расширена. Модель *eERM* — расширенная модель «сущность - отношение» как раз и представляет собой расширение классической модели Чена. Вместо ее полного названия часто используется аббревиатура eERM.

Данная модель играет существенную роль при описании данных в архитектуре ARIS. Модель данных eERM используется для создания информационных моделей, отражающих структуру информации, которая обрабатывается в бизнес-процессах организации.

Объекты модели eERM приведены в табл. 21, а возможные связи — в табл. 22. Используемая при конструировании панель инструментов показана на рис. 115.

*Тип сущности* используется для представления материальных и нематериальных типов объектов.

*Тип отношений* необходим для представления устойчивых типов отношений между типами объектов при моделировании.

*Кластер* является совокупностью некоторого количества связанных типов сущностей и предназначен для представления сложных объектов, а также для указания структуры информации, изменение которой фиксируется в событии.

Объект *обобщение* предназначен для указания операции *обобщения/специализации* некоторых типов объектов. Обобщаемый тип сущности присоединяется к вершине треугольника, обобщающие/специализирующие типы сущностей присоединяются к основанию треугольника.



Рис. 115. Панель инструментов для создания моделей eERM

## Стандартный и расширенный методологические фильтры. Обзор моделей

Таблица 21. Объекты eERM-модели

	Английское название объекта	Русское название объекта
	Connection	Связь
	Entity type	Тип сущности
	Relationship type	Тип отношений
	Reinterpretation relationship type	Переопределенный тип отношений
	Cluster	Кластер
	Generalization	Обобщение
	Generalization	Обобщение
	Generalization	Обобщение
	Generalization	Обобщение
	D attribute (ERM)	Описательный атрибут (ERM)
	K attribute (ERM)	Ключевой атрибут (ERM)
	EK attribute (ERM)	Ключевой наследуемый атрибут (ERM)
	ERM domain	ERM-домен
	Attribute type group	Группа типов атрибутов
	Enumeration	Перечисление
	Unit	Элемент
	Unit number	Номер элемента
	Enumeration list occurrence	Экземпляр списка

*Описательный атрибут* предназначен для описания свойств моделируемых объектов.

*Ключевой атрибут* служит для указания свойства объекта, значение которого однозначно определяет экземпляр типа объекта реального мира. Ключевые атрибуты могут быть только у сущностей. Связи не могут иметь ключевых атрибутов. Ключевой атрибут может быть простым или

составным. Простой атрибут состоит из одного атрибута. Составной состоит из нескольких атрибутов, т. е. только совокупность значений этих атрибутов однозначно определяет экземпляр типа объекта реального мира. ARIS не имеет средств для поддержки составного идентификатора.

ERM-домен используется для указания множества значений атрибута, т.е. его области значений. С помощью одного домена могут быть определены несколько атрибутов.

*Группа типов атрибутов* предназначена для группировки атрибутов по некоторому признаку. Она может служить для указания составного идентификатора.

Объект *перечисление* применяется для указания значения атрибута, когда область его значений не соответствует наименованию атрибута, например, для уточнения смысла таких атрибутов, как *характер, тип производства, тип руководства*.

Опираясь на ряд различных подходов к расширению модели «сущность-отношение», можно выделить четыре основных оператора проектирования: *классификацию, обобщение, агрегацию и группировку*.

Операторы *проектирования* обеспечивают формальную поддержку процесса создания

модели данных. Их применение гарантирует систематический подход и дает возможность тому, кто знает существующую структуру данных, понять суть процесса проектирования. Анализ условий выполнения бизнес-процессов с точки зрения их структур данных помогает разработчикам структурировать известные условия, базируясь на новом представлении, а также создавать новые отношения, не рассматривавшиеся до сих пор.

*Классификация.* При помощи классификации объекты (сущности) одного и того же типа идентифицируются и ассоциируются в соответствии с некоторым признаком (типом сущностей). Один объект идентичен другому, если он описан теми же свойствами (атрибутами).

Таблица 22. Связи объектов в модели eERM

Английское название связи	Русское название связи
Belongs to	Принадлежит
Defines	Определяет
Differentiates according to value of	Дифференцируется в зависимости от
Has	Имеет
Has link to	Имеет связь с
Is attribute type group	Является группой типов атрибутов для
Is describing for	Описывает
Is primary key for	Является первичным ключом для
Is subtype of	Является подтипом
Is supertype of	Является супертипом

### Стандартный и расширенный методологические фильтры. Обзор моделей

*Обобщение/Специализация.* При обобщении аналогичные типы объектов группируются под одним, более старшим типом объекта. Как показано на рис. 116а, тип сущности *клиент* и тип сущности *поставщик* объединены под общей концепцией «Участник процесса». Свойства, описанные атрибутами и являющиеся общими для обоих исходных объектов, присваиваются обобщенному объекту. Таким образом, остается описать только такие атрибуты, которые отличны от атрибутов исходных типов объектов. Образование нового типа сущности *участник бизнес-процесса* представляется графически в виде треугольника, который также является отношением и означает «есть».

Под *специализацией* понимается разделение некоторого общего множества (например, объектов) на подмножества.

Оператор специализации является инверсным по отношению к оператору обобщения (пример: сущность *участник процесса* разделяется на сущности *клиент* и *поставщик*). Специализированные объекты наследуют свойства обобщенных объектов. Кроме наследования, специализированные типы объектов могут обладать также собственными атрибутами. Графически специализация и обобщение представляются одинаково. По этой причине соединения на рисунке не отображаются стрелками, указывающими направление.

Специализация в первую очередь поддерживает подход к структуре данных «сверху вниз», обобщение же используется при подходе «снизу вверх».

В рамках специализации полнота и степень разделяемости на поднаборы могут определяться в процессе их создания.



Рис. 116. Обобщение/Специализация (а) и полная специализация (б)

О неразделяемых подмножествах будем говорить в том случае, когда экземпляр одного объекта может быть частью обоих подмножеств.

Для приведенного выше примера это означает, что некоторый клиент в то же самое время может быть и поставщиком. Если экземпляр ассоциируется только с одним подмножеством, то множества являются разделяемыми.

Когда все специализированные типы объектов, возможные для одного критерия специализации, входят в состав одного обобщенного типа объекта, говорят о полной специализации. Например, тип сущности *человек* может быть разделен на типы сущностей *мужчина* и *женщина* (рис. 116б). Специализация полностью определена, если в качестве критерия специализации выбран «пол».

Комбинация этих критериев приводит в результате к следующим четырем случаям, которые выделяются, чтобы более точно определить оператор *обобщение/специализация*:

- раздельная/полная;
- раздельная/неполная;
- нераздельная/полная;
- нераздельная/неполная.

*Агрегация.* Агрегация описывает формирование нового типа объекта с помощью комбинации существующих типов объектов. В этом контексте новый тип объекта может нести новые свойства.

В модели eERM агрегация представляется как формирование типов отношений (рис. 117). Агрегация типов сущностей *заказ на производство* и *маршрутизация* создает новый объект *маршрутизация заказа*.



Рис.117. Пример агрегации

Оператор агрегации также применяется к отношениям. В этом случае существующий тип отношения, называемый переопределенным, трактуется как тип сущности и, таким образом,

может стать отправной точкой для создания другого, нового отношения (рис. 118).



Рис. 118. Агрегация с переопределением типа отношений

### Стандартный и расширенный методологические фильтры. Обзор моделей

Первая агрегация используется для формирования типа отношения *маршрутизация заказа* из типов сущностей *заказ на производство* и *маршрутизация*. Ключевые атрибуты *№ заказа* и *№ маршрута* образуют сложный ключевой атрибут для отношения к переопределенному типу отношений *маршрутизация заказа*. Теперь многочисленные операции могут быть связаны непосредственно с ним. Следовательно, отношение *операция над заказом* формируется между типами отношений *маршрутизация заказа* и типом сущности *операция*. Поскольку отношения могут создаваться только между типами сущностей, необходимо, чтобы первоначальный тип отношения *маршрутизация заказа* был интерпретирован как тип сущности. На рис. 118 это иллюстрируется с помощью ромба, заключенного в прямоугольник.

Этот переопределенный тип отношения трактуется как нормальный тип сущности. Процедура создания типа отношения из нескольких существующих типов сущностей отображается с помощью линий, направленных к вершинам ромба. Однако линии, означающие новые отношения по переопределенным типам отношений, подходят к краям прямоугольника, окружающим ромб, а не к самому ромбу (рис. 118).

Несмотря на то, что сложный ключ можно заменить более простым, целесообразнее использовать сложные ключи, поскольку они позволяют лучше отслеживать процесс создания модели данных.

В модели eERM сложный структурный элемент разделяется таким образом, чтобы образовать прозрачную структуру. Для лучшего понимания и для того, чтобы не нарушить стройность концепции, вводят сложные объекты в виде кластеров данных.

Кластер данных описывает логическую модель некоторых типов сущностей и отношений в модели данных. Это необходимо для описания сложных объектов. Кластеры данных содержат не только типы сущностей и отношений, но и другие кластеры данных. В отличие от типов сущностей и отношений кластеры данных могут быть легко встроены в иерархию, что позволяет в процессе создания модели данных поддерживать подход «сверху вниз». Использование кластеров данных может быть полезно и при объединении моделей в ходе проектирования «снизу вверх».

*Группировка.* В процессе группировки формируются группы, составленные из элементов некоторого множества сущностей.

В примере на рис. 119 все составные части сущности *оборудование* объединяются в *группу оборудования*. Группа оборудования — это независимый объект, который может быть описан более точно с помощью дополнительных атрибутов (наименование группы оборудования, число деталей оборудования), которые не характеризуют отдельные части оборудования. Другим примером может служить группировка рабочих мест в отделы или объединение элементов, связанных с обработкой заказа, в группу *заказ*



Рис. 119. Группировка

В модели eERM используется расширенное понятие мощности отношений. При ссылке на число, характеризующее мощность отношений, ранее упоминались только верхние пределы допустимого числа экземпляров отношений.

Мощности на рис. 120а указывают, что проект может быть связан с максимальным числом ( $w$ ) служащих и один служащий может участвовать в максимальном числе  $\{n\}$  проектов.

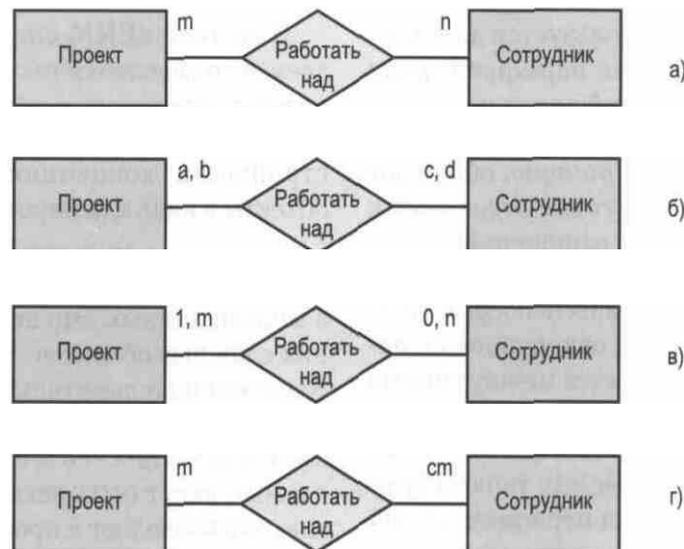


Рис. 120. Верхнее/нижнее ограничение мощности отношений

Кроме верхнего ограничения, может также представлять интерес нижнее ограничение. Это имеет место в том случае, когда нужно специфицировать минимальное число экземпляров отношений. Для этих целей мощность отношений может быть выражена двумя буквами. Буквы (a, b) на рис. 120б указывают, что для каждого проекта можно определить по крайней мере a и не более чем b экземпляров отношений типа *работать над*. Это означает, что каждый проект может выполняться по крайней мере a и не более чем b сотрудниками. Буквы (c, d) указывают, что один сотрудник может участвовать по крайней мере в c и (не более чем) в d проектах.

Таким образом, каждое отношение выражается двумя степенями сложности {min — минимальной и max — максимальной}. Нижним границам часто присваиваются значения 0 или 1. Диапазон значений верхней границы определяется как  $1 < \max < *$  (где \* — универсальный знак).

Нижняя граница  $\min = 0$  указывает, что сущность может участвовать в одном отношении, но это необязательно. Нижняя граница  $\min = 1$  указывает, что сущность должна участвовать по крайней мере в одном отношении.

Нижние границы на рис. 120в указывают, что сотрудник может участвовать в проекте, но это необязательно ( $\min = 0$ ). Этим здесь подчеркивается, что могут быть сотрудники, которые не участвуют ни в одном проекте, в то время как проект должен выполняться по крайней мере одним сотрудником ( $\min = 1$ ). И наоборот, по крайней мере по одному сотруднику должно быть прикреплено к каждому проекту.

Если разрешены минимальное значение 0 или 1 и максимальные значения 1 или \*, то для пары {min, max} возможны следующие четыре сочетания: (1,1), (1, m), (0,1) и (0,t). Приведенные ниже сокращения являются обычными для указанных сочетаний:

- 1 (соответствует (1,1)),
- c (соответствует (0,1)),
- m (соответствует (1, w)),
- cm (соответствует (0,m)), где c = choice (выбор) и m == multiple (многократный).

На рис. 120г представлен пример с использованием указанной нотации.

Благодаря расширению понятия мощности посредством определения нижней и верхней границ может быть рассмотрена зависимость между объектами данных. По определению,

типы отношений и типы переопределенных отношений существуют в силу существования типов сущностей, связанных с ними. Следовательно, они не существуют в изоляции. Это означает, что типы отношений зависимы от других типов сущностей как с точки зрения существования, так и в плане идентификации.

Кроме того, имеют место типы сущностей, которые в действительности обладают единственным ключевым атрибутом, но при этом зависят от наличия других сущностей.

Эти типы зависимостей могут появляться, например, при групповых операциях. Как показано на рис. 121, сущность *отдел* имеет смысл только в том случае, если он содержит по крайней мере одну сущность *рабочее место*. В свою очередь *рабочее место* только тогда имеет смысл, когда оно входит в *отдел*. Эти зависимости существования выражаются степенью сложности, или мощностью. В нотации (*min, max*) они определяются как (1,1) и (1,\*).



Рис. 121. Зависимость существования

Определение зависимости существования в модели данных обеспечивает целостность основных данных, что является важным условием при реализации. Его выполнение гарантирует, что целостность содержимого базы данных обеспечивается даже после выполнения некоторых транзакций. В приведенном выше примере удалить отдел можно только в том случае, если будут удалены все рабочие места, входящие в данный отдел.

## 7.2.2. Диаграмма атрибутов eERM-модели — eERM Attribute allocation diagram

Элементы диаграммы атрибутов eERM-модели приведены в табл. 23. Связи в данной диаграмме такие же, как и в модели eERM. Панель инструментов, используемая при конструировании диаграммы, показана на рис. 122.



Рис. 122. Панель инструментов для создания диаграммы атрибутов eERM-модели

Таблица 23. Объекты в диаграмме атрибутов eERM-модели

	Английское название объекта	Русское название объекта
	Connection	Связь
	D attribute (ERM)	Описательный атрибут (ERM)
	K attribute (ERM)	Ключевой атрибут (ERM)
	EK attribute (ERM)	Ключевой наследуемый атрибут (ERM)
	ERM domain	ERM-домен
	Generalization	Обобщение
	Entity type	Тип сущности
	Relationship type	Тип отношений
	Reinterpretation relationship type	Переопределенный тип отношений
	Attribute type group	Группа типов атрибутов
	Enumeration	Перечисление
	Unit	Элемент
	Unit number	Номер элемента
	Enumeration list occurrence	Экземпляр списка

Модели данных eERM, которые отображают только типы сущностей и типы отношений, очень часто имеют довольно сложную структуру. Диаграммы, включающие атрибуты сущностей и отношений, теряют наглядность.

С помощью диаграммы атрибутов eERM-модели можно описать атрибуты для каждого типа сущности и отношения на отдельной диаграмме. В нее можно включить тип

объекта из диаграммы eERM (тип сущности или тип отношения) в виде копии экземпляра. Таким образом, может быть смоделировано распределение атрибутов по объектам eERM. В этом контексте можно различать, является ли атрибут, связанный с объектом eERM, ключевым атрибутом, внешним ключом или описательным атрибутом. На рис. 123 приведен соответствующий пример.

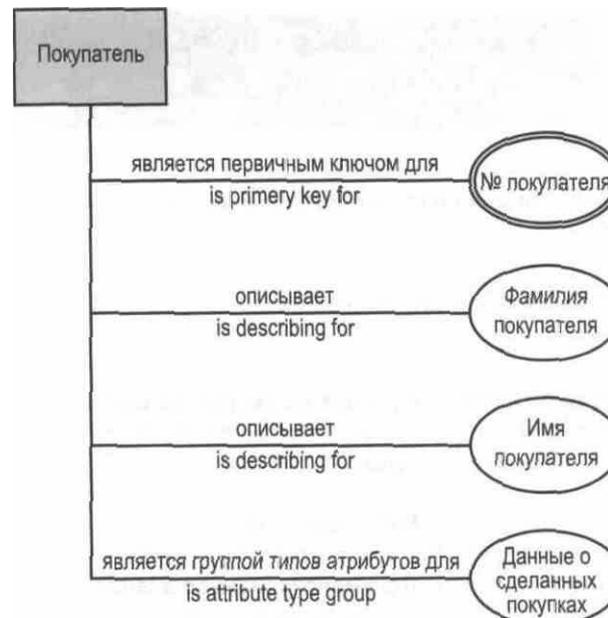


Рис. 123. Диаграмма атрибутов eERM-модели для сущности «Покупатель»

Кроме представления и распределения отдельных атрибутов для объектов eERM-модели, в этом типе диаграммы можно также отобразить группу типов атрибутов (например, группа «Данные о сделанных покупках» на рис. 123) и их распределение. Группа типов атрибутов представляет группу атрибутов одного типа сущности из диаграммы eERM, которые семантически близко связаны. Это позволяет создать группу атрибутов, содержащую все атрибуты объектов eERM-модели, которые вместе образуют, например, вторичный ключ.

### 7.2.3. Диаграмма структуры знаний — Knowledge structure diagram

Данная диаграмма входит в расширенный методологический фильтр и предназначена для моделирования процесса управления знаниями. Для этой же цели служит *карта знаний*, которая будет рассмотрена в разделе 7.3.4.

Диаграмма структуры знаний относится к уровню описания требований. Карта знаний, подобно детализирующим типам моделей,

помещается на уровень описания требований моделей процессов/управления.

Объекты диаграммы структуры знаний приведены в табл. 24. Возможные связи между объектами представлены в табл. 25. Панель инструментов, используемая при создании модели, показана на рис. 124.

Целью управления знаниями является систематическое обновление знаний как ресурса организации, важность которого постоянно возрастает. Термин «управление знаниями» охватывает разработку, мониторинг, поддержку и совершенствование стратегий, процессов, организационных структур и технологий для эффективной обработки знаний. Он включает также все действия, имеющие отношение к приобретению знаний, их подготовке, передаче и использованию. Действия по управлению знаниями обычно не совершаются в изоляции. Они сопровождаются главным образом оперативными и директивными бизнес-процессами в компании. Следовательно, необходим интегральный взгляд на бизнес-процессы, обработку знаний, организационные структуры, информационные системы и т.д.



Рис. 124. Панель инструментов для создания диаграммы структуры знаний

Таблица 24. Объекты диаграммы структуры знаний

	Английское название объекта	Русское название объекта
	Connection	Связь
	Knowledge category	Категория знаний
	Documented knowledge	Документированное знание
	Log	Запись
	Card file	Картотека
	File	Файл
	Document	Документ
	Expertise	Эксперт
	Folder	Папка
	Microfiche	Микрофиша
	Telephone	Телефон
	Fax	Факс
	Cluster	Кластер
	Information object	Информационный объект
	Entity type	Тип сущности
	Relationship type	Тип отношения
	Package	Пакет
	Class	Класс
	Object	Объект
	Application system type	Тип прикладной системы
	Application system class	Класс прикладной системы

## Стандартный и расширенный методологические фильтры. Обзор моделей

Таблица 25. Связи в диаграмме структуры знаний

Английское название связи	Русское название связи
Is documented in	Документировано в
Is generalization of	Является обобщением
Is managed with	Управляется с помощью
Subsumes	Содержит

Большинство из этих аспектов может быть отражено с помощью таких известных методов ARIS, как eEPC-диаграммы, организационные схемы, диаграммы окружения функций, eERM-модели и другие. Однако если цель заключается в точном представлении, анализе и совершенствовании дисциплины управления знаниями, необходимы дополнительные средства представления для идентификации и структурирования содержимого соответствующих категорий знаний, описания и распределения знаний в пределах организации, моделирования создания и использования знаний в бизнес-процессах.

В связи с этим добавляются два новых типа объектов — *категория знаний* и *документированные знания*. Кроме того, существующие типы моделей для представления бизнес-процессов (eEPC, PCD, производственный и офисный процессы и т. п.) расширяются конструкциями для управления созданием и использованием знаний.

Объект *категория знаний* описывает конкретные знания в той или иной области.

Примерами являются знание об управлении проектом, конкретные знания о технологии производства, знания о клиенте и конкурентах, знания о системах менеджмента качества и т.д. Эти объекты помогают классифицировать знания, которыми обладает организация или которые ей необходимы.

Знания, распределяемые по отдельным категориям, могут быть:

- неявными, т.е. такими, которые невозможно полностью документировать;
- явными, т.е. такими, которые могут быть документированы в виде описаний или технической документации (чертежей);
- знаниями о квалификации служащих или их групп.

Категории знаний часто содержат более одного вида знаний. Например, знания об управлении проектом включают опыт руководства проектом и рекомендации по управлению проектом.

Приведенные в табл. 26 дополнительные атрибуты, так же как и основные атрибуты *описание*, *комментарий*, *источник* и т.д., служат для более подробного описания категорий знаний.

**Моделирование бизнеса. Методология ARIS**

**Таблица 26. Атрибуты объекта «Категория знаний»**

Наименование атрибута	Диапазон значений	Описание/Пример
Частота корректировки	Градации: ежечасно, ежедневно, еженедельно, ежемесячно, ежегодно, редко, никогда	Частота корректировки описывает, как часто знания отдельной категории должны обновляться, чтобы отражать текущее состояние. Например, основные тригонометрические знания должны редко обновляться, если вообще должны, в то время как знания о ценах на товар необходимо обновлять ежедневно или ежечасно.
Важность	Процент: 0-100	Категория "Важность" может изменяться от 0 (совсем неважно) до 100% (чрезвычайно важно).
Степень охвата	Процент: 0-100	Текущая степень охвата для отдельных знаний может меняться от 0 (совсем не охвачено) до 100% (максимально возможный охват).
Преимущество	Процент: 0-100	Относительное преимущество компании над конкурентами в терминах знаний может изменяться от 0 (конкуренты имеют максимально возможное преимущество над компанией) до 100% (компания имеет максимально возможное преимущество над конкурентами).
Использование знаний	Процент: 0-100	Степень применения отдельной категории знаний может изменяться от 0 (конкретные знания не используются совсем) до 100% (оптимальное использование конкретных знаний).
Желаемая степень охвата	Процент: 0-100	Желаемая степень охвата отдельных знаний может изменяться от 0 (совсем не охвачены) до 100% (максимально возможный охват).
Важность в будущем	Градации: резкое снижение, снижение, постоянная, возрастание, резкое возрастание	Важность в будущем отражает ожидаемую тенденцию в изменении степени важности данной категории знаний для компании.
Скорость структурных изменений	Процент: 0-100	Скорость структурных изменений отражает, насколько быстро должны изменяться методы, направленные на приобретение конкретных знаний (0%: не требуется изменений, 100%: максимальная скорость структурных изменений).

**Стандартный и расширенный методологические фильтры. Обзор моделей**

Атрибуты, приведенные в табл. 26, могут быть использованы как основа при оценке уровня управления знаниями. Часто оказывается полезным графическое представление их величин.

В отличие от объекта *категория знаний*, который описывает неявные и явные знания, тип объекта *документированные знания* включает только знания, которые явно документированы или которые в принципе могут быть документированы. Примером такого типа знаний является знание об

использовании программного обеспечения, которое документировано в руководстве.

При распределении информации по категориям выявление различий между категориями знаний и документированными знаниями позволяет оценить возможности и ограничения информационных систем, если только документированные знания могут быть сохранены, переданы или обработаны электронным способом. Объект *документированные знания* имеет такие же типы атрибутов, что и объект *категория знаний*.

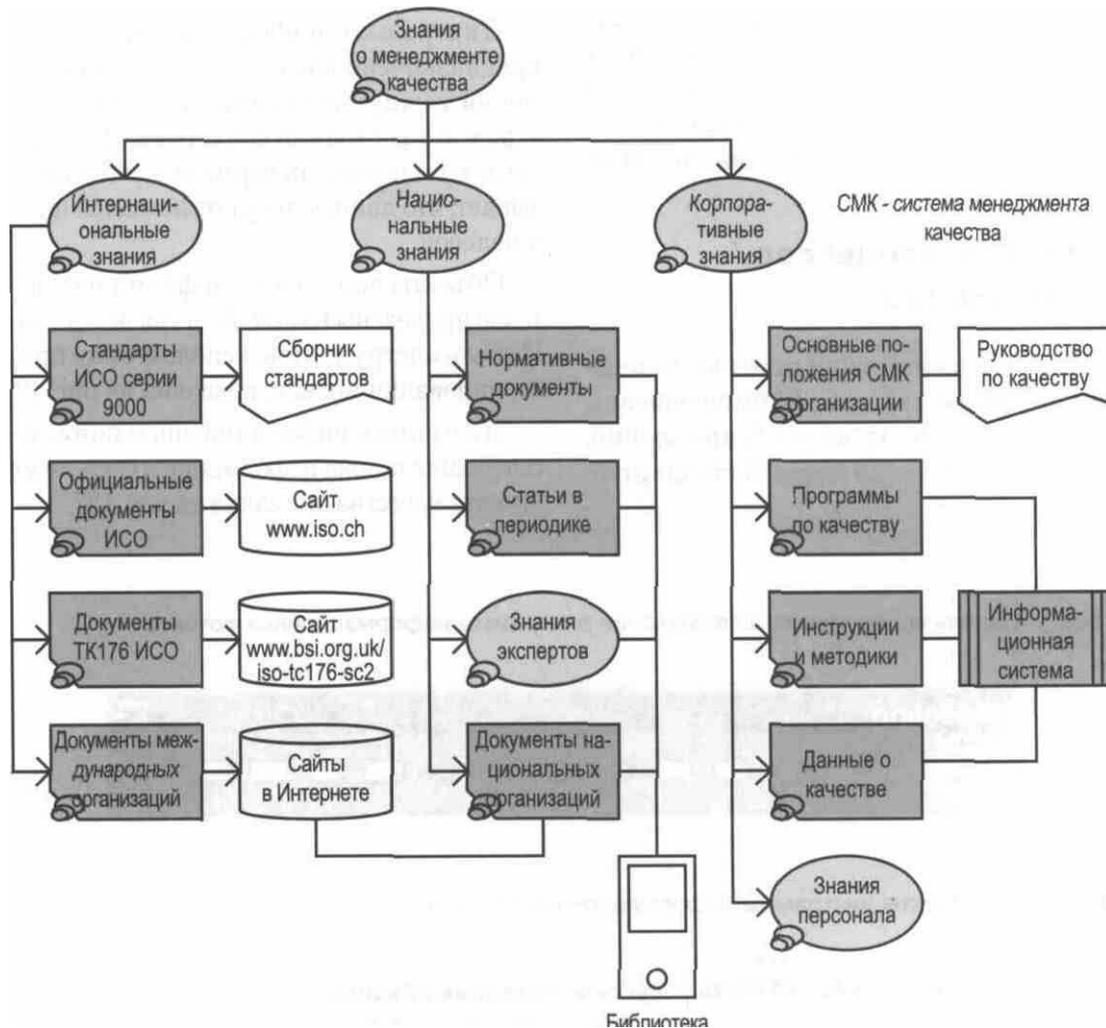


Рис. 125. Диаграмма структуры знаний, описывающая «Знания о менеджменте качествах»

**Моделирование бизнеса. Методология ARIS**

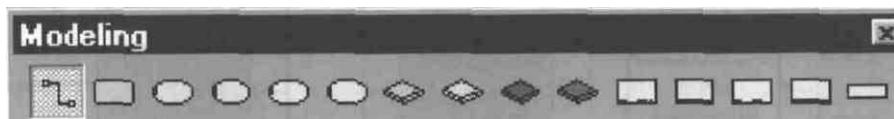
С помощью диаграммы структуры знаний пользователь может поместить категории знаний в подгруппы, основываясь на их содержанием. Пример диаграммы структуры знаний приведен на рис. 125. Категория знаний может включать другие категории знаний, а также документированные знания. Документированные знания можно разделить на несколько подкатегорий документированных знаний, но они не могут включать общие знания.

Для документированных знаний диаграмма структуры знаний может отображать информационную среду, где знания документированы, или указывать, какие информационные объекты в модели данных (или классах объектно-ориентированных систем) служат для документирования этих знаний. Наконец, могут быть также смоделированы типы или классы прикладных систем для управления знаниями.

### 7.3. Модели процессов/управления

Ранее при рассмотрении простого методологического фильтра (глава 6) были описаны следующие модели процесса/управления, также входящие в состав моделей стандартного фильтра:

**Рис. 126.** Панель инструментов для создания диаграммы информационных потоков



**Таблица 27.** Объекты диаграммы информационных потоков

	Английское название объекта	Русское название объекта
	Connection	Связь
	Function	Функция
	System organization unit type	Тип системной организационной единицы

- событийная цепочка процесса (раздел 6.4.1);
- диаграмма окружения функции (раздел 6.4.2);
- производственный и офисный процессы (раздел 6.4.3);
- диаграмма цепочки добавленного качества (раздел 6.4.4).

#### 7.3.1. Диаграмма информационных потоков — Information flow diagram

Диаграмма информационных потоков предназначена для описания потоков информации между функциями. Для этой цели в диаграмме необходимо связать две функции посредством потока информации. Эта связь указывает, что данные текут от исходной функции к целевой.

Объекты диаграммы информационных потоков приведены в табл. 27, а связи — в табл. 28. Панель инструментов, используемая при конструировании модели, показана на рис. 126.

Диаграмма информационных потоков, описывающая потоки информации в системе менеджмента качества, показана на рис. 127.

## Стандартный и расширенный методологические фильтры. Обзор моделей

Продолжение табл. 27

	Английское название объекта	Русское название объекта
	System organization unit	Системная организационная единица
	Organizational unit type	Тип организационной единицы
	Organizational unit	Организационная единица
	SAP function	Функция SAP
	SAP function	Функция SAP
	SAP function	Функция SAP
	SAP function	Функция SAP
	System organization unit type	Тип системной организационной единицы
	System organization unit	Системная организационная единица
	Organizational unit type	Тип организационной единицы
	Organizational unit	Организационная единица
	Information flow	Информационный поток

Таблица 28. Связи объектов в диаграмме информационных потоков

Английское название связи	Русское название связи
Can be assigned to	Может быть связан(а) с
Executes	Выполняет
Information flow	Информационный поток
Is assigned to	Связан(а)с
Is IT responsible for	Отвечает по IT за
Is executed by	Выполняется
Is received from	Направляется к
Is sent from	Направляется от
Is technically responsible for	Отвечает за техническую часть
Receives	Получает от
Sends	Направляет в

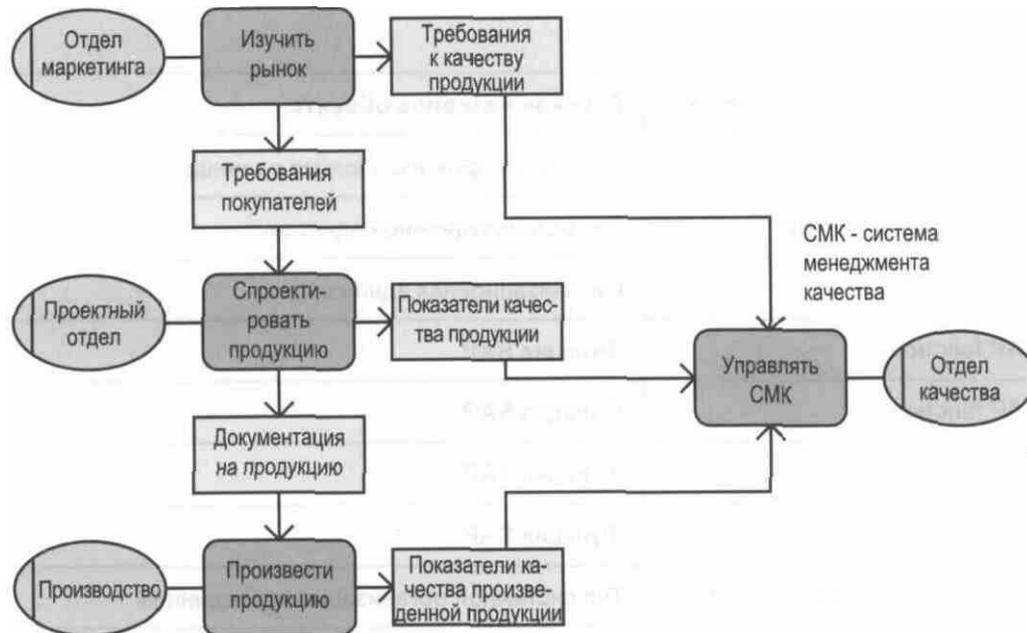


Рис. 127. Диаграмма информационных потоков

### 7.3.2. Матрица выбора процессов — Process selection matrix

Матрица выбора процессов отображает различные сценарии выполнения процесса при по

мощи привязки основных процессов к отдельным сценариям. Изображения объектов матрицы выбора процессов показаны на рис. 128.

Допустимые типы связей между объектами диаграммы:

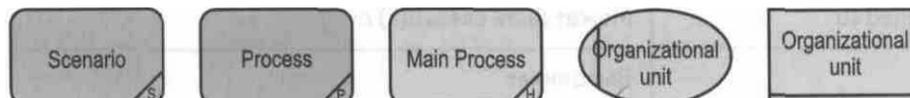


Рис. 128. Изображения объектов матрицы выбора процессов: (слева направо) сценарий, процесс, главный процесс, организационная единица, организационная единица

- *consist of* — состоит из;
- *is assigned to* — связан(а) с.

Матрицы выбора процессов позволяют определить, какие функции, «проигранные» в сценариях процесса, могут иметь место на предприятии. По этой причине в диаграмму должны быть включены все главные функции (функции сценария) прикладной системы или функции модели-прототипа производства.

При моделировании матрицы выбора процессов используются следующие типы графических символов (рис. 128):

- *сценарий* — некоторый вариант выполнения процесса, объединяющий главные процессы в группы. На рис. 129 группы отделены пунктирными линиями;
- *главный процесс* — главные функции в функциональном дереве, с которыми

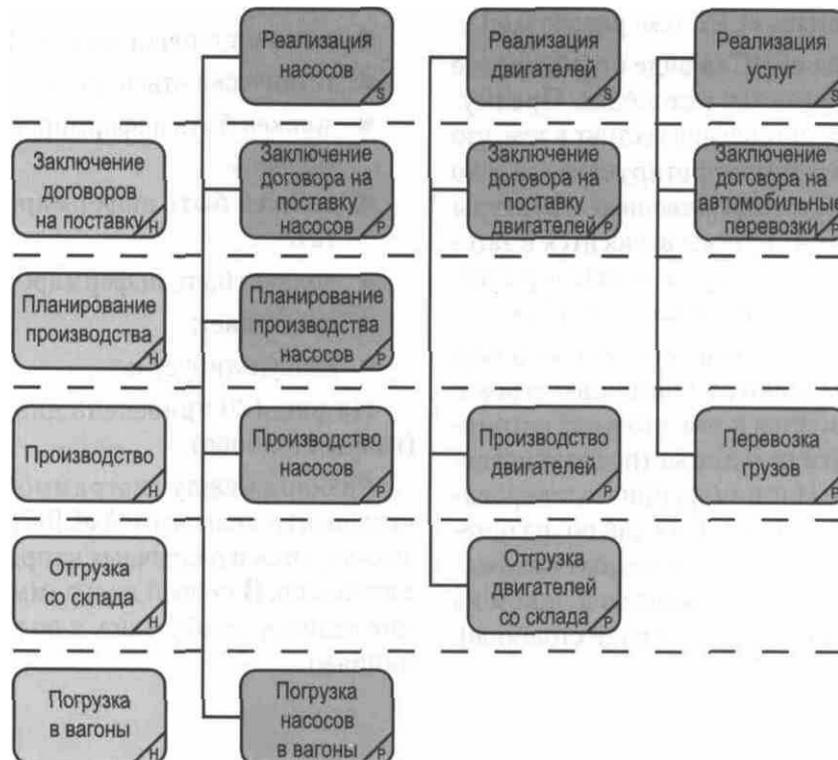
**Стандартный и расширенный методологические фильтры. Обзор моделей**

связаны процессы (функции из сценария процессов);

*процесс* — представляет функции из сценария процесса, которые более подробно описаны в модели-прототипе с помощью моделей процесса.

Пример матрицы выбора процесса приведен на рис. 129.

Матрицы выбора процессов предприятия формируются на основе информации, содержащейся в:



**Рис.129. Матрица выбора процесса**

- модели организационной структуры и моделях внешних субъектов;
- деревьях продуктов/услуг.

В матрицы выбора процессов включаются как процессы, необходимые для получения основных внешних выходов предприятия, так и процессы производства внутренних продуктов в одних подразделениях, для использования их другими подразделениями.

### 7.3.3 Диаграмма eEPC (в виде столбцов) — eEPC (column display)

Данная диаграмма входит в расширенный методологический фильтр.

Изображение объектов диаграммы eEPC (в виде столбцов) совпадает с изображением объектов диаграммы eEPC (см. раздел 6.4.1).

В диаграммах eEPC (в виде столбцов) все символы располагаются в столбцах. Преимущество такого представления состоит в том, что eEPC-диаграмма интерпретируется гораздо проще. Элементы организационной структуры и информационной системы выносятся в заголовок диаграммы. Все другие символы размещаются во второй строке каждого столбца.

Отличительная особенность всех моделей, которые представляются с помощью строк и столбцов, заключается в том, что в них автоматически создаются невидимые (подразумеваемые) отношения. Например, при моделировании прикладных систем и функций подразумеваемое отношение *поддерживают* автоматически создается по

умолчанию в одном из столбцов диаграммы eEPC (в виде столбцов).

Отношение *выполняет* будет создано автоматически между элементами организационной структуры и функциями. Можно ввести дополнительные столбцы, названия которых будут соответствовать подразумеваемым отношениям. Например:

- вносит вклад в ...;
- решает;
- отвечает с точки зрения ИТ за ...;
- технически отвечает за ...;
- должен быть информирован в случае отмены ...;
- должен быть информирован о результате ...;
- должен быть информирован о ...;
- принимает;
- консультирует по...

На рис. 130 приведена диаграмма eEPC (в виде столбцов).

Разница между диаграммой eEPC (в виде столбцов) и диаграммой eEPC (в виде строки) заключается в различных направлениях моделирования. В первой диаграмме моделирование ведется сверху вниз, а во второй — слева направо.

Стандартный и расширенный методологические фильтры. Обзор моделей

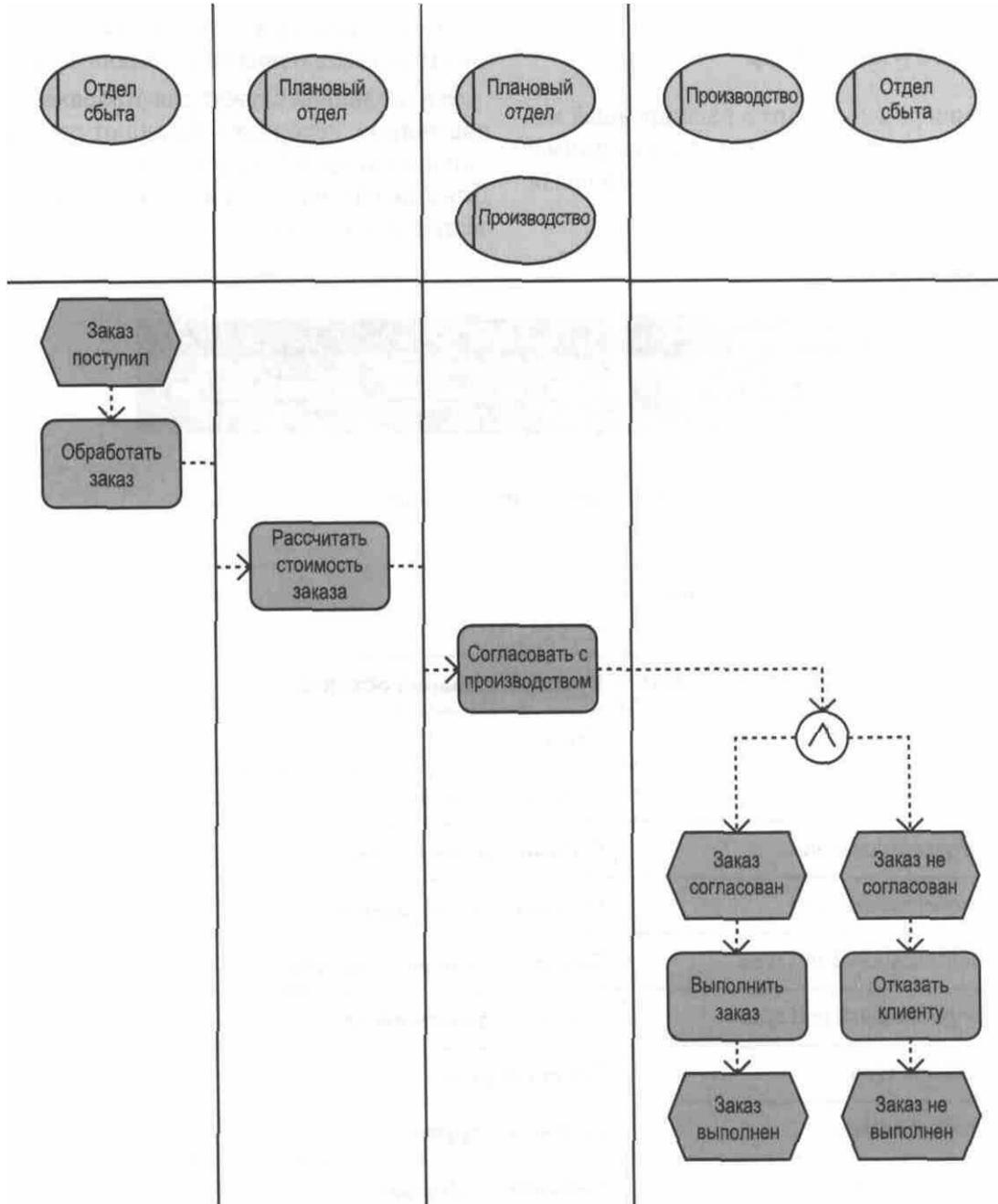


Рис. 130. Диаграмма eEPC (в виде столбцов)

### Моделирование бизнеса. Методология ARIS

#### 7.3.4. Карта знаний — Knowledge map

Эта диаграмма входит в расширенный методологический фильтр. Объекты диаграммы и их названия приведены в табл. 29, а связи между объектами — в табл. 30.

Панель инструментов, используемая при конструировании модели, показана на рис. 131.

Карты знаний служат для отображения типов знаний, которыми обладают служащие или организационные единицы компании. Для каждого типа знаний можно ввести количественную оценку.

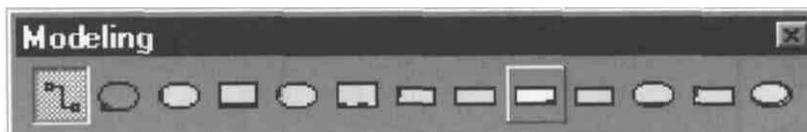


Рис. 131. Панель инструментов для создания карты знаний

Таблица 29. Объекты карты знаний

	Английское название объекта	Русское название объекта
	Connection	Связь
	Knowledge category	Категория знания
	Organizational unit	Организационная единица
	Organizational unit	Организационная единица
	Organizational unit type	Тип организационной единицы
	Organizational unit type	Тип организационной единицы
	Position type	Тип должности
	Internal person	Штатный сотрудник
	External person	Внештатный сотрудник
	Position	Должность
	Location	Местоположение
	Person type	Тип сотрудника
	Group	Группа

## Стандартный и расширенный методологические фильтры. Обзор моделей

Таблица 30. Связи объектов в карте знаний

Английское название связи	Русское название связи
Has knowledge of	Владеет знанием
Requires	Требует

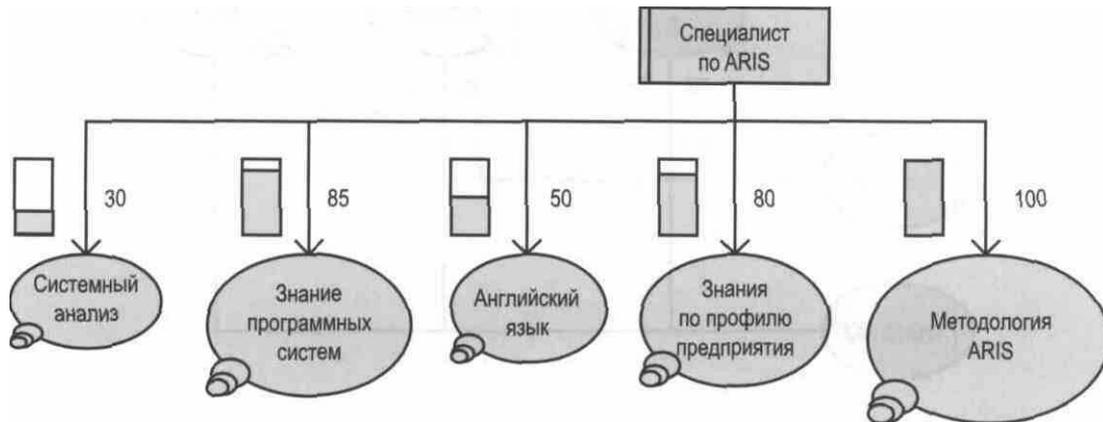


Рис 132. Карта знаний, отражающая квалификацию специалиста в области ARIS

Карта знаний отображает распределение различных категорий знаний в рамках организации. Типы объектов в организационной схеме (например, *организационная единица*, *должность*, *сотрудник*, *оборудование*, *группа*) могут быть привязаны к категориям знаний с помощью соединения *требует* {requires}. Кроме факта, что отдельный сотрудник или организационная единица обладает знаниями конкретной категории, может быть также описана и степень охвата.

Соединение *требует* содержит атрибут *степень охвата* {degree of coverage}, который может иметь процентное выражение степени охвата знаний в категории знаний, выбранной для организационной единицы. Значение 100% соответствует максимальному охвату, а значение 0% означает полное отсутствие знаний, что соответствует отсутствию соединения.

Кроме количественной оценки, возможна качественная оценка в виде графика. Для этой цели служит атрибут соединения *качество охвата* {coverage quality}, который может принимать значения *низкий*, *средний*, *высокий* и *максимальный*. Эта информация может быть представлена с помощью графического символа на соединении, как показано на рис. 132. На рисунке также показано альтернативное визуальное представление для различных степеней охвата: объекты *категории знаний* изображаются графическими фигурами разного размера.

Значения атрибутов *степень охвата* и *качество охвата* не являются взаимосвязанными. Если используются оба атрибута, то предполагается, что значение *низкий* относится к степени охвата <25%, *средний* — 25-50%, *высокий* — 50-75%, *максимальный* — 75-100%.

Карта знаний может быть ориентирована на организационную структуру, т.е. соответствующая категория знаний «связывается» с каждой организационной единицей.

Для карт знаний часто используется матричное представление. Это может быть

достигнуто организацией нескольких экземпляров одной и той же категории знаний в столбцы, как показано на рис. 133. В этом примере приведены имена категории знаний, расположенные в верхней части рисунка, как это принято обычно в таблице.

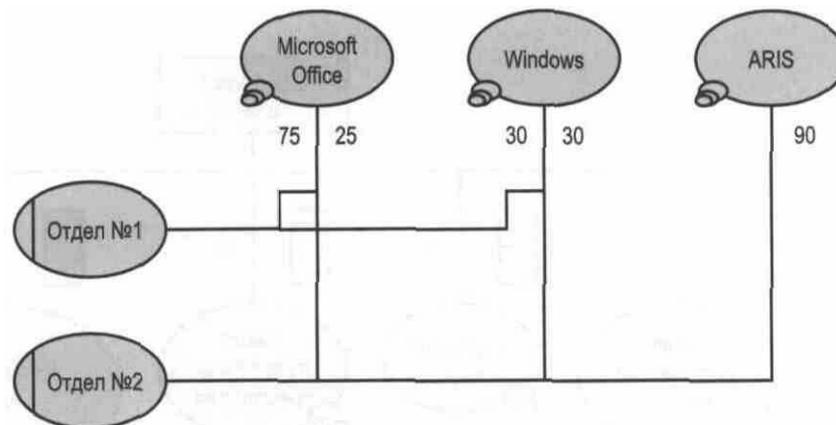


Рис. 133. Карта знаний: матричное представление

Создание и применение знаний в бизнес-процессах компании моделируются с помощью моделей, используемых для представления бизнес-процессов (eEPC, eEPC с потоком материалов, офисный процесс, производственный процесс, PCD, PCD с потоком материалов). Типы объектов категория знаний и документированные знания

доступны и в указанных диаграммах. Можно определить, какой тип знаний необходим для выполнения функции, и отметить, какие знания создаются и/или документируются, когда функция выполняется (рис. 134).

## Стандартный и расширенный методологические фильтры. Обзор моделей

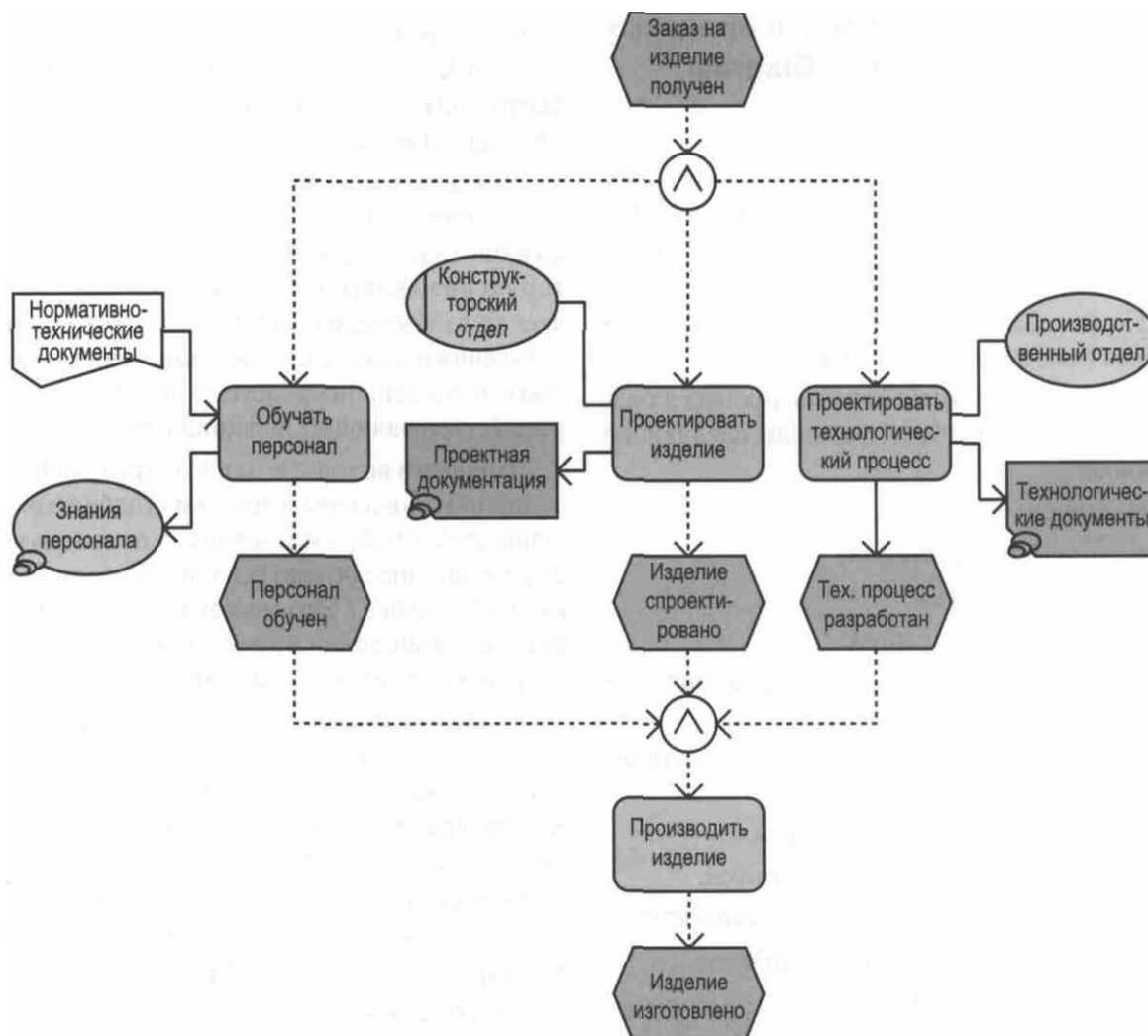


Рис. 134. Отражение знаний в моделях eERP

Этот тип представления позволяет изучать бизнес-процессы в терминах обработки знаний. Например, может быть обнаружен пробел в необходимых знаниях и определен

уровень повышения квалификации, который требуется при выполнении отдельной функции.

### 7.3.5. Диаграмма цепочки процесса — Process Chain Diagram (PCD)

Данная диаграмма включена только в расширенный методологический фильтр. Нотации объектов моделей диаграмм PCD и eEPC совпадают (см. раздел 6.4.1).

На рис. 135 приведена процесс обработки заказа в виде диаграммы PCD.

PCD — диаграмма цепочки процесса разделена на ряд столбцов, имеющих следующие названия:

- event — событие;
- function — функция;
- data — данные;
- medium — носитель;
- application system — прикладная система;
- organizational unit — организационная единица;
- screen/list — экран/список;
- batch — пакетная обработка;
- dialog — интерактивная обработка;
- manual — ручная обработка;
- objects — объект;
- product/service — продукт/услуга;
- objective — цель;
- other — прочее.

В этих столбцах располагаются соответствующие объекты, тем самым диаграмма приобретает четкую структуру, что делает ее легко читаемой. Отдельные столбцы могут быть опущены. Заголовки также можно изменить.

Первый и второй столбцы отображают логическую последовательность выполнения процесса. Отдельные функции процесса, представленные во втором столбце, связаны с событиями, которые инициируют их выполнение или переключение между ними. Функции и события связаны пунктирными стрелками, отображающими поток управления функциями.

Отношение между событиями и функциями составляет процедурную последовательность функций как логическую цепочку событий, которую называют *цепочкой процесса*. Логическая взаимозависимость возможных точек ветвления и циклов потока управления может быть выражена посредством логических операций, связывающих функции и события.

Входные и выходные данные, требующиеся функциям, показаны в третьем столбце в виде сущностей. Отображаться могут не только информационные объекты, но и носители (четвертый столбец). Это может быть документ, список, написанная вручную квитанция или устройство памяти (например, жесткий диск).

В пятом столбце могут быть показаны задействованные в моделируемом процессе прикладные информационные системы, например, электронная почта, Интранет, Интернет и другие, а также их отдельные компоненты.

Организационные единицы, ответственные за выполнение отдельных функций моделируемого процесса, показаны в шестом столбце.

Столбец *экран/список* может содержать объекты, характеризующие представление информации в ходе выполнения процесса.

Столбцы *пакетная обработка*, *интерактивная обработка*, *ручная обработка* и столбец *прикладная система* представляют дополнительную информацию о степени использования информационных технологий для поддержки отдельной функции процесса.

В остальных столбцах может содержаться дополнительная информация, характеризующая цели процесса и его отдельных функций, а также задействованные и производимые продукты и услуги.

Стандартный и расширенный методологические фильтры. Обзор моделей

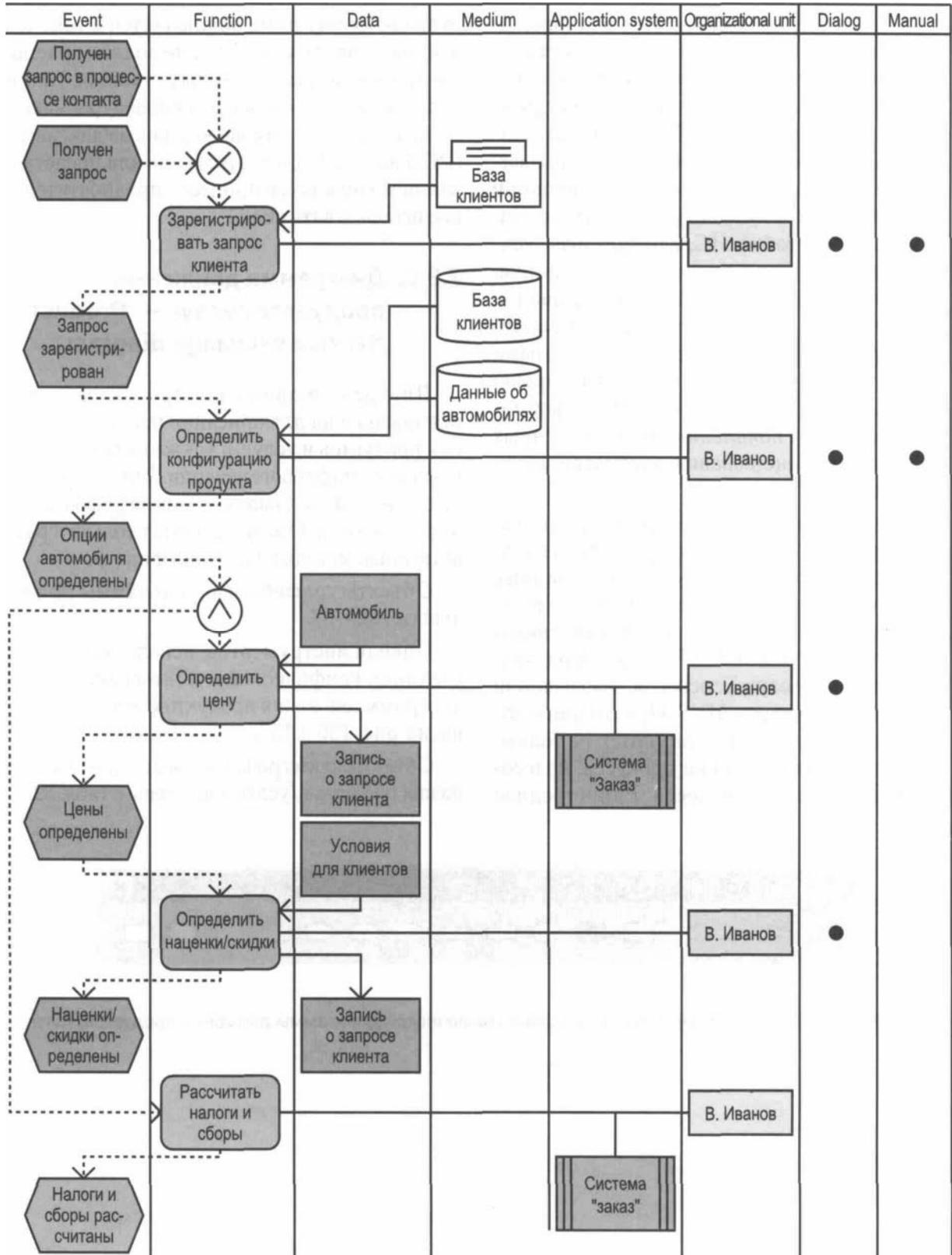


Рис. 135. PCD-диаграмма цепочки процесса

При анализе процесса, который отображает некоторую реальную ситуацию, недостатки его организации или причины неэффективности могут быть определены с помощью диаграммы цепочки процесса. Такими недостатками могут быть информационные разрывы, когда последовательные («соседние») функции выполняются с помощью различных, не связанных между собой, ИС, или организационные разрывы, когда, например, сотрудники, участвующие в выполнении данной функции, относятся к различным структурным единицам. Кроме того, становятся четко видимыми лишние входы и выходы, что говорит о процедурной избыточности данных. Эта информация приводит к появлению многочисленных идей по совершенствованию рассматриваемого процесса.

При описании на верхнем уровне диаграммы процессов создаются с относительно высокой степенью обобщения (без детализации). Поскольку эти диаграммы используются главным образом для отображения взаимосвязей отдельных компонент ARIS (моделей различных типов), они служат основой для создания управляющей модели ARIS. При создании управляющей модели используются не только PCD-диаграммы цепочки процесса, но и событийные цепочки процесса. Единственное отличие между ними заключается в том, что

элементы диаграмм eEPC не должны располагаться в предопределенных столбцах. Таким образом, модель небольшой процедуры может быть построена с помощью одной из диаграмм (PCD или eEPC), в то время как для представления модели всего процесса предпочтительнее использовать eEPC.

### 7.3.6. Диаграмма движения продуктов/услуг — Product/ Service exchange diagram

Диаграмма движения продуктов/услуг предназначена для описания процесса создания продуктов и услуг, а также их передвижения в масштабах организации. Эта диаграмма существует в двух видах — пиктографическом и графическом. Оба вида входят только в расширенный методологический фильтр.

Объекты графической диаграммы приведены в табл.31.

Панели инструментов, используемые при создании графической и пиктографической диаграмм движения продуктов/услуг, показаны на рис.136 и 137.

Объекты пиктографической диаграммы движения продуктов/услуг приведены в табл. 32.

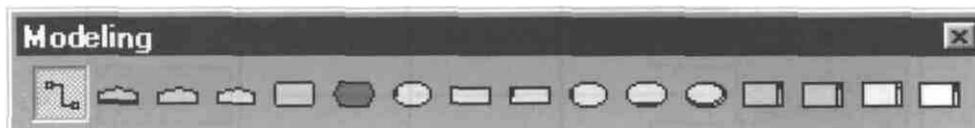


Рис. 136. Панель инструментов для создания графической диаграммы движения продуктов/услуг

## Стандартный и расширенный методологические фильтры. Обзор моделей

Таблица 31. Объекты графической диаграммы движения продуктов/услуг

	Английское название объекта	Русское название объекта
	Connection	Связь
	Product/Service	Продукт/услуга
	Service	Услуга
	Information service	Информационная услуга
	Function	Функция
	Event	Событие
	Organizational unit type	Тип организационной единицы
	Internal person	Штатный сотрудник
	Person type	Тип сотрудника
	Organizational unit	Организационная единица
	Location	Местоположение
	Group	Группа
	Operating resource type	Тип операционного ресурса
	Technical operating supply type	Тип вспомогательных технических средств
	Material type	Тип материала
	Packaging material type	Тип упаковочного материала



Рис. 137. Панель инструментов для создания пиктографической диаграммы движения продуктов/ услуг

Таблица 32. Объекты в пиктографической диаграмме движения продуктов/услуг

	Английское название объекта	Русское название объекта
	Connection	Связь
	Product	Продукт
	Service	Услуга
	Information service	Информационная услуга
	Function (Office)	Функция (офисный процесс)
	Function (Manufacturing)	Функция (производственный процесс)
	Event	Событие
	Organizational unit type	Тип организационной единицы
	Person (m)	Мужчина
	Person (f)	Женщина
	Person type	Тип сотрудника
	Organizational unit	Организационная единица
	Location	Местоположение
	Location	Местоположение
	Group	Группа
	Supplier	Поставщик
	Customer	Клиент
	Company	Компания
	Machine	Машина
	Robot	Робот
	Tool	Инструмент
	Material type	Тип материала
	Material type	Тип материала
	Packaging material	Упаковочный материал

### Стандартный и расширенный методологические фильтры. Обзор моделей

Диаграмма движения продуктов/услуг предназначена для представления процесса их создания (рис.138), а также обмена ими внутри компании. Продукт может быть:

- материалом;
- операционным ресурсом;

• вспомогательным техническим средством;

- упаковочным материалом.

Продукты/услуги могут являться входами/выходами функций, а также иметь связь с начальными и конечными событиями для этих функций.

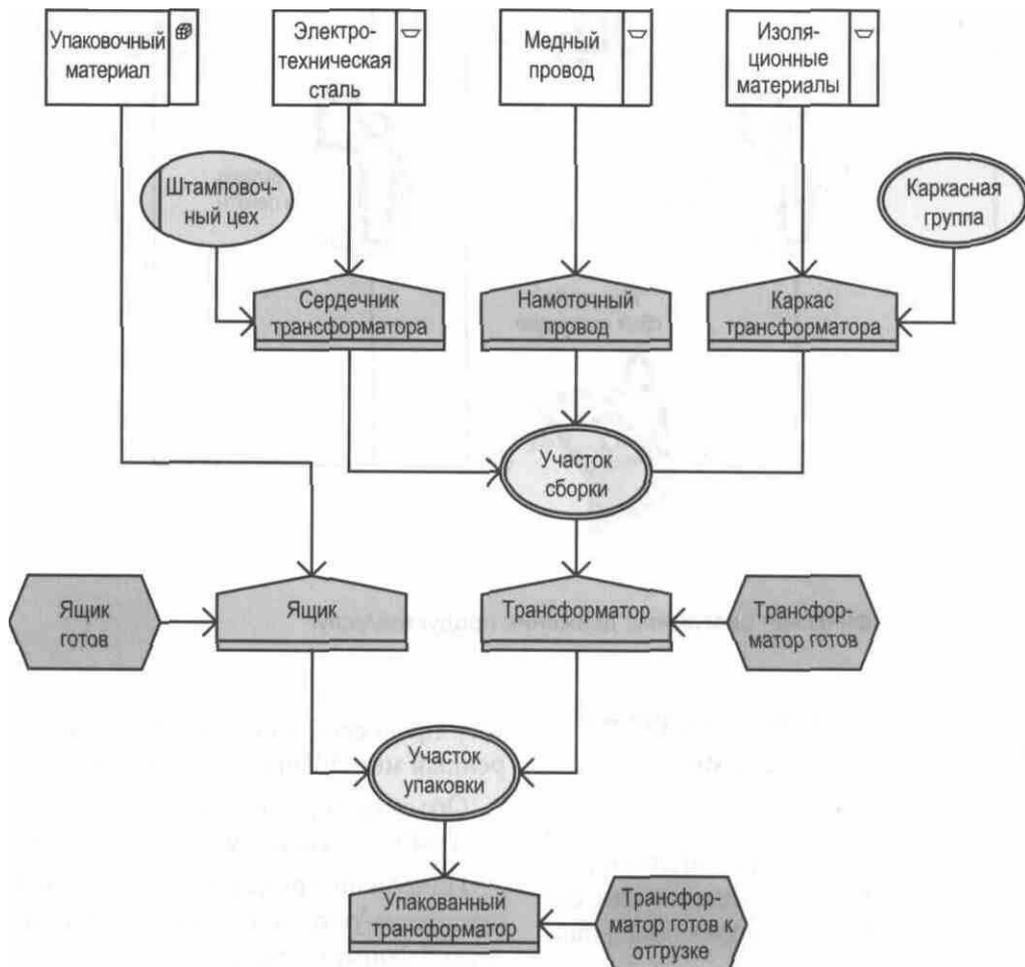


Рис. 138. Графическая диаграмма движения продуктов/услуг, моделирующая процесс производства трансформатора

Пиктографическая диаграмма движения продуктов/услуг — более наглядная и может использоваться

менее квалифицированными пользователями ARIS. Пример такой диаграммы приведен на рис. 139.

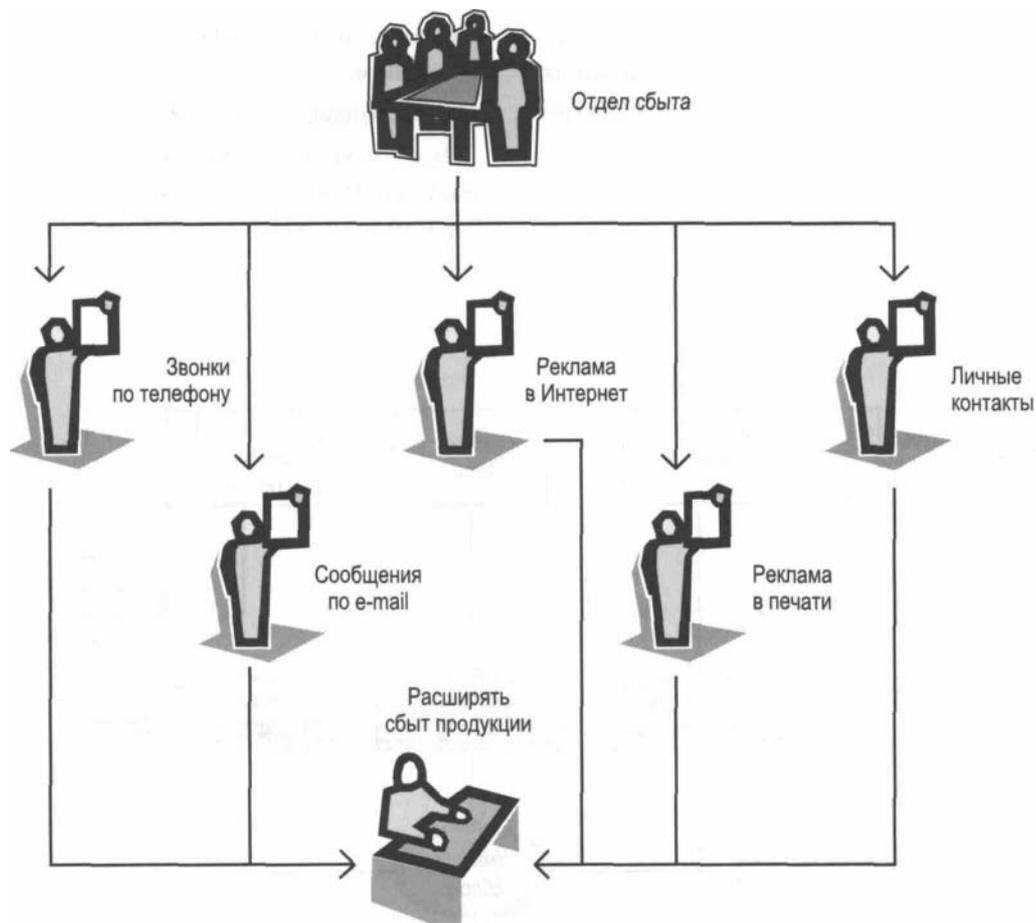


Рис. 139. Пиктографическая диаграмма движения продуктов/услуг

### 7.3.7. Дерево продуктов/услуг — Product/Service tree

Дерево продуктов/услуг предназначено для представления продукции в виде составляющих ее частей. Эта диаграмма также существует в двух формах — графической и пиктографической. Она

входит только в расширенный методологический фильтр.

Объекты дерева продуктов/услуг приведены в табл. 33, а связи между ними — в табл. 34.

Панели инструментов для создания дерева продуктов/услуг в графической и пиктографической формах приведены на рис. 140 и 141.

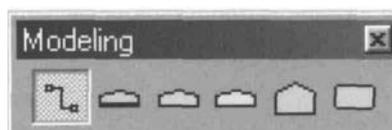


Рис. 140. Панель инструментов для создания дерева продуктов/услуг в графической форме

## Стандартный и расширенный методологические фильтры. Обзор моделей

Таблица 33. Объекты дерева продуктов/услуг в графической форме

	Английское название объекта	Русское название объекта
	Connection	Связь
	Product/Service	Продукт/услуга
	Service	Услуга
	Information service	Информационная услуга
	Objective	Цель
	Function	Функция

Таблица 34. Связи объектов дерева продуктов/услуг

Английское название связи	Русское название связи
Can replace	Может заменять
Has output of	Имеет на выход
Is input for	Является входом для
Supports	Поддерживает
Subsumes	Содержит

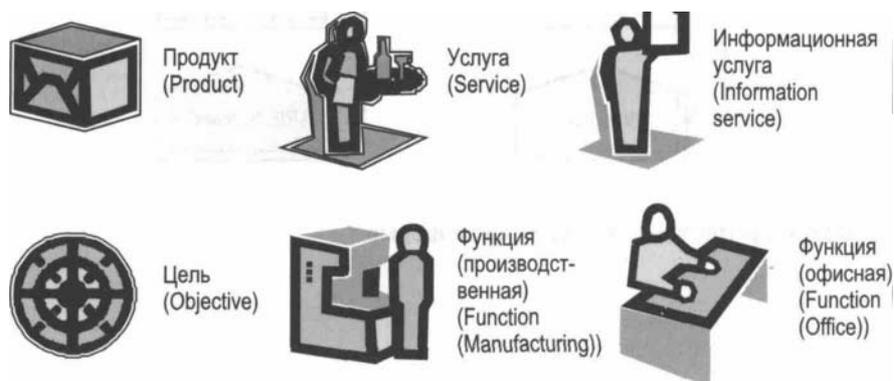


Рис. 141. Объекты дерева продуктов/услуг в пиктографической форме

### Моделирование бизнеса. Методология ARIS

Продукты/услуги, являющиеся результатом деятельности организации, иногда целесообразно представить в виде набора (моделей) составляющих их компонент. Диаграмма дерева продуктов/услуг позволяет описать эти взаимосвязи, а каждая из компонент может быть описана отдельной моделью. В дереве продуктов/услуг могут

быть представлены отношения замещения результатов, когда используется (потенциальный) заменяющий продукт или услуга.

Два одинаковых по содержанию примера дерева продуктов/услуг приведены на рис. 142 (графическая форма) и на рис. 143 (пиктографическая форма).

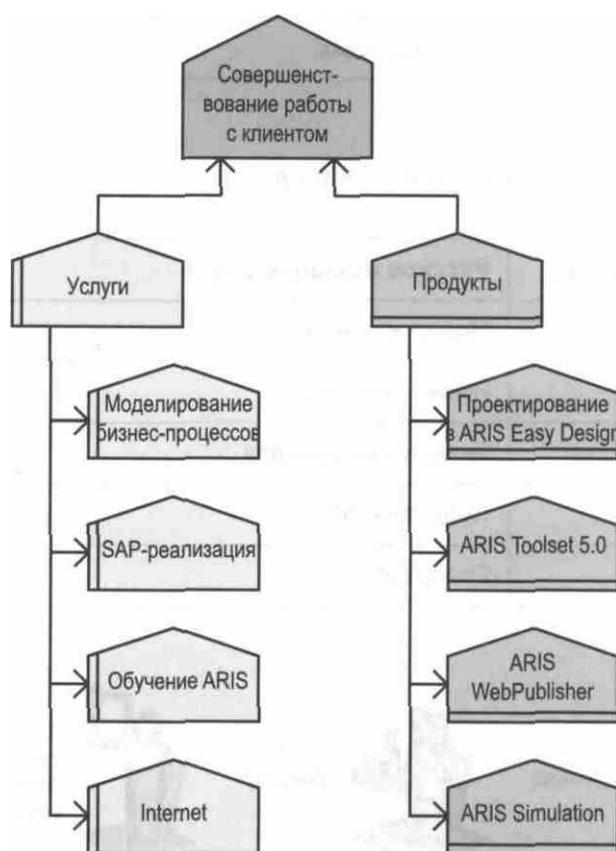


Рис. 142. Дерево продуктов/услуг в графической форме

## Стандартный и расширенный методологические фильтры. Обзор моделей

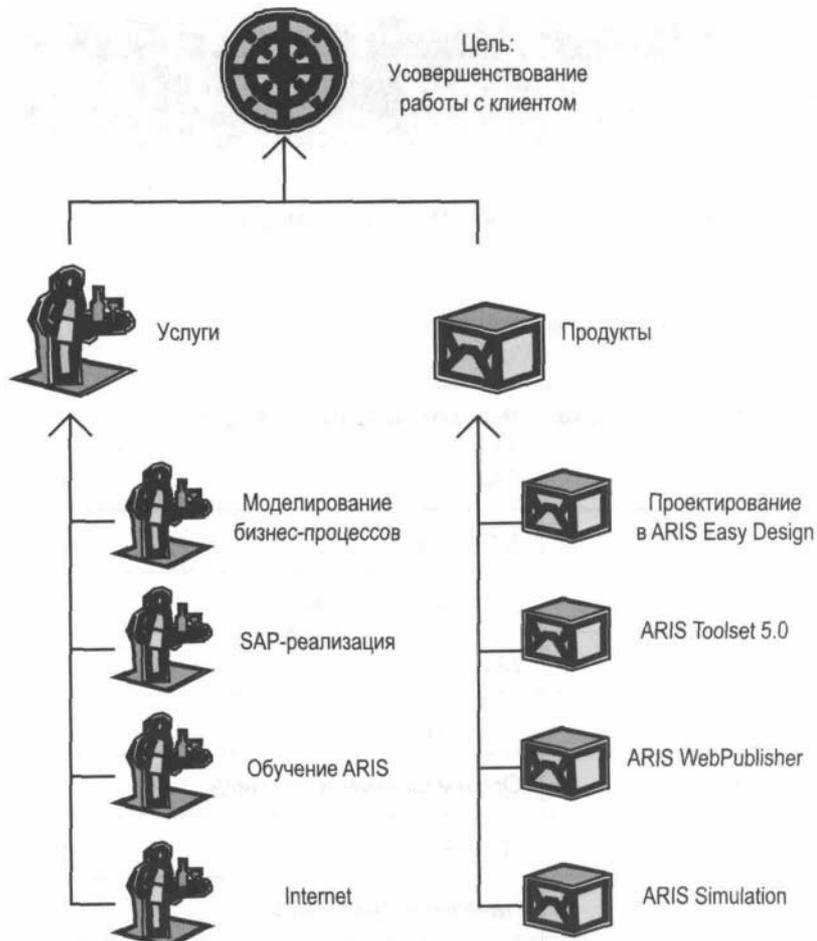


Рис. 143. Дерево продуктов/услуг в пиктографической форме

### 7.3.8. UML-диаграмма действий — UML Activity diagram

В ARIS включен ряд моделей, построенных с использованием объектно-ориентированного унифицированного языка моделирования (UML — *Unified Modeling Language*) в стандартизированной версии рабочей группы OMG (*Object Management Group*). UML базируется на объектно-ориентированных подходах OMT, Booch и OOSE.

UML-диаграмма действий описывает процесс как последовательность действий. В языке UML действия всегда относятся к объектам. Таким образом, диаграммы действий привязываются к классу, операции или использованию приложений и описывают относящийся к ним внутренний процесс.

Объекты UML-диаграммы действий приведены в табл. 35, а связи между ними — в табл. 36. Панель инструментов, используемая при создании моделей, показана на рис. 144.

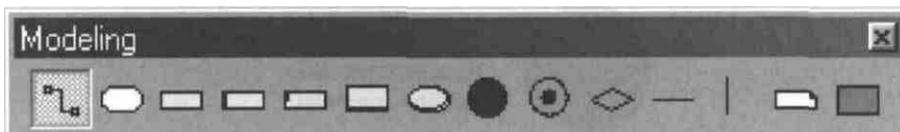


Рис. 144. Панель инструментов для создания UML-диаграммы действий

Таблица 35. Объекты UML-диаграммы действий

	Английское название объекта	Русское название объекта
	Connectivity	Связь
	Activity	Действие
	Internal person	Штатный сотрудник
	Position	Должность
	Person type	Тип сотрудника
	Organizational unit	Организационная единица
	Group	Группа
	Initial state	Начальное состояние
	Final state	Конечное состояние
	Decision	Решение
	Split/Synchro	Разделить/синхронизировать
	Split/Synchro	Разделить/синхронизировать
	Note	Примечание
	Object state	Состояние объекта

## Стандартный и расширенный методологические фильтры. Обзор моделей

Таблица 36. Связи объектов в UML-диаграмме действий

Английское название связи	Русское название связи
Activates	Активизирует
Belongs to	Принадлежит
Executes	Выполняет
Has output of	Имеет на выходе
Is input for	Является входом для
Is predecessor of	Предшествует
Leads to	Порождает событие через
Links	Предшествует

UML-диаграмма действий предназначена для описания процессов. Каждый описываемый процесс должен начинаться с объекта *начальное состояние* и завершаться объектом *конечное состояние*. Объект *действие* представляет состояние с каким-то внутренним действием и одним или несколькими переходами. Переходы обозначаются соединениями, соответствующими отношениям между объектами *действие*. Они могут иметь простые отношения с другими *действиями*, а также множественные входящие и выходящие отношения.

Множественные внешние отношения могут быть сформулированы как условия. Для их представления используется *объект решение*. Моделирование условий с помощью этого объекта не является обязательным. В качестве альтернативы пользователи могут ввести несколько внешних связей. На диаграмме рекомендуется показывать условие в атрибуте *связь* для отношений *порождает событие через* и *активизирует*.

Объект *разделить/синхронизировать* может быть использован для активизации в одно и то же время нескольких последовательно

соединенных объектов *действие* или одного объекта *действие*, зависящего от переходов в рамках нескольких предыдущих *действий*.

*Действия* могут предполагать особое начальное состояние объекта и создавать особое конечное состояние объекта. Состояния объектов изображаются типом объекта *состояние объекта*, который описывается соединениями *является входом для* или *имеет на выходе* в виде отношения с *действиями*.

UML-диаграмма отображает организационную ответственность за выполнение *действий* с помощью так называемого *плавающего коридора*. Плавающий коридор — это столбец, где перечислены все действия, за которые ответственна организационная единица. Для этой цели UML-диаграмма действий содержит предопределенную таблицу с двумя линиями. Ответственная организационная единица (внутренний участник, место, тип участника и т.д.) располагается на верхней линии, в то время как нижняя линия отводится для объектов *действие*, *решение*, *разделить/синхронизировать*, *состояние объекта* и *примечание*.

На рис. 145 представлена UML-диаграмма действий с соответствующими компонентами.

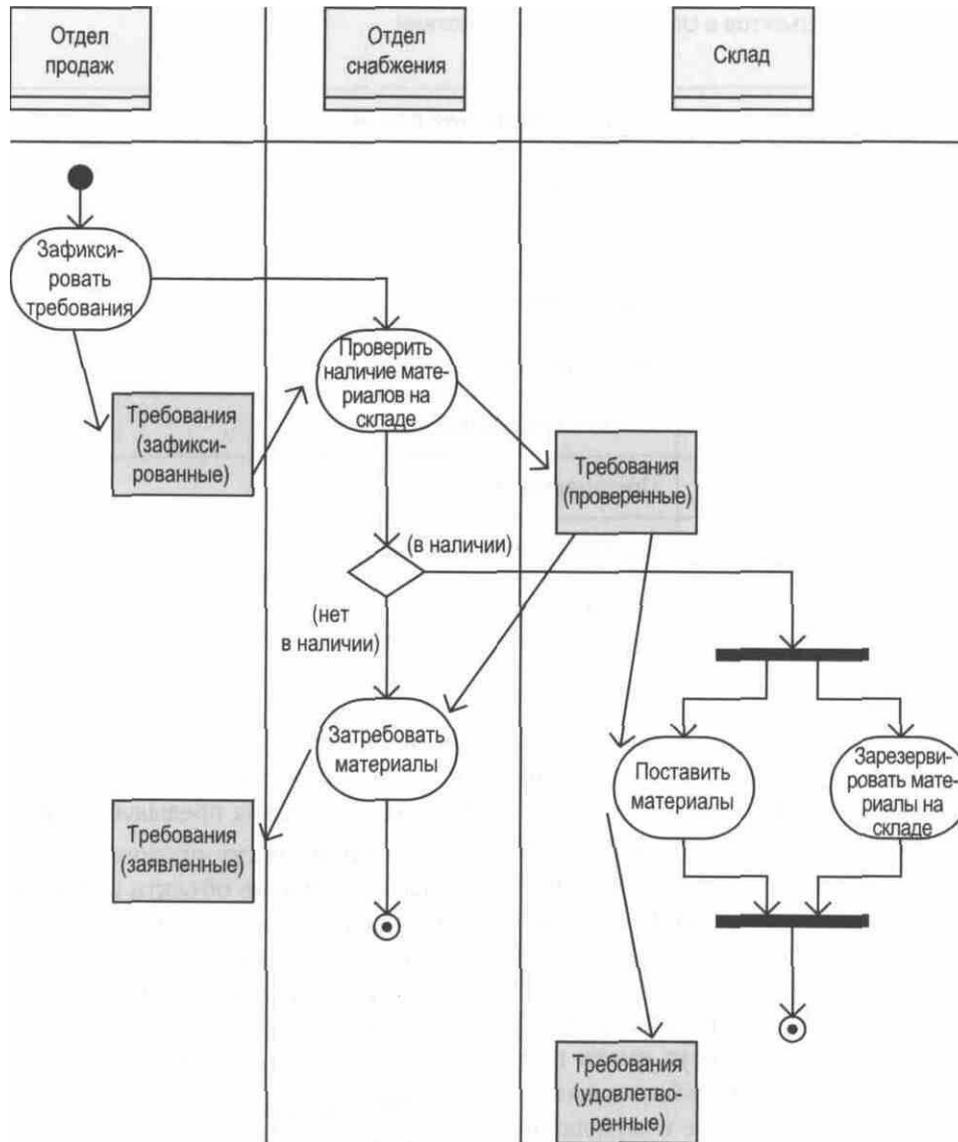


Рис. 145. UML-диаграмма действий

### 7.3.9. UML-диаграмма класса — UML Class diagram

UML-диаграмма класса отражает статические отношения между такими объектами и модели, как *класс*, *объект* и *интерфейс*.

Панель инструментов, используемая при конструировании моделей, показана на рис. 146. Объекты и их названия приведены в табл. 37.

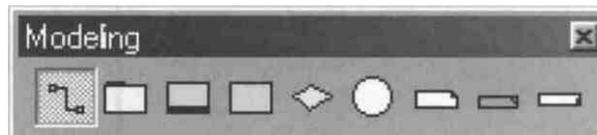


Рис. 146. Панель инструментов для создания UML-диаграммы класса

Таблица 37. Объекты UML-диаграммы класса

	Английское название объекта	Русское название объекта
	Connection	Связь
	Package	Пакет
	Class	Класс
	Object	Объект
	Association	Ассоциация
	Interface	Интерфейс
	Note	Примечание
	Operation	Операция
	Attributes	Атрибуты

Число связей между объектами данной диаграммы довольно велико, и с ними можно ознакомиться в соответствующих методологических разделах руководства ARIS.

UML-диаграмма класса определяет классы, к которым с помощью отношения *имеет в своем составе* могут быть привязаны соответствующие операции (методы) и атрибуты.

Отношения, которые классы имеют друг с другом, моделируются в UML-диаграммах

класса с помощью соединения *ассоциируется с*. Соединения осуществляются непосредственно между классами для бинарных отношений. Ассоциация, изображаемая в виде ромба, представляет собой группу связей, одинаковых по структуре и семантике. Если ассоциация является классом, может быть использовано соединение *представляет свойства для*. Множество соединений *ассоциируется с* может быть введено в атрибуты *множество (источник)* и *множество (цель)*.

В UML агрегация и композиция отражают специальные отношения *ассоциируется с*. Они описываются с помощью входящего в атрибут *тип агрегации* соединения *ассоциироваться* и изображаются небольшим

белым (агрегация) или черным (композиция) ромбом на конце соединения *ассоциироваться*. Пример представлен на рис. 147

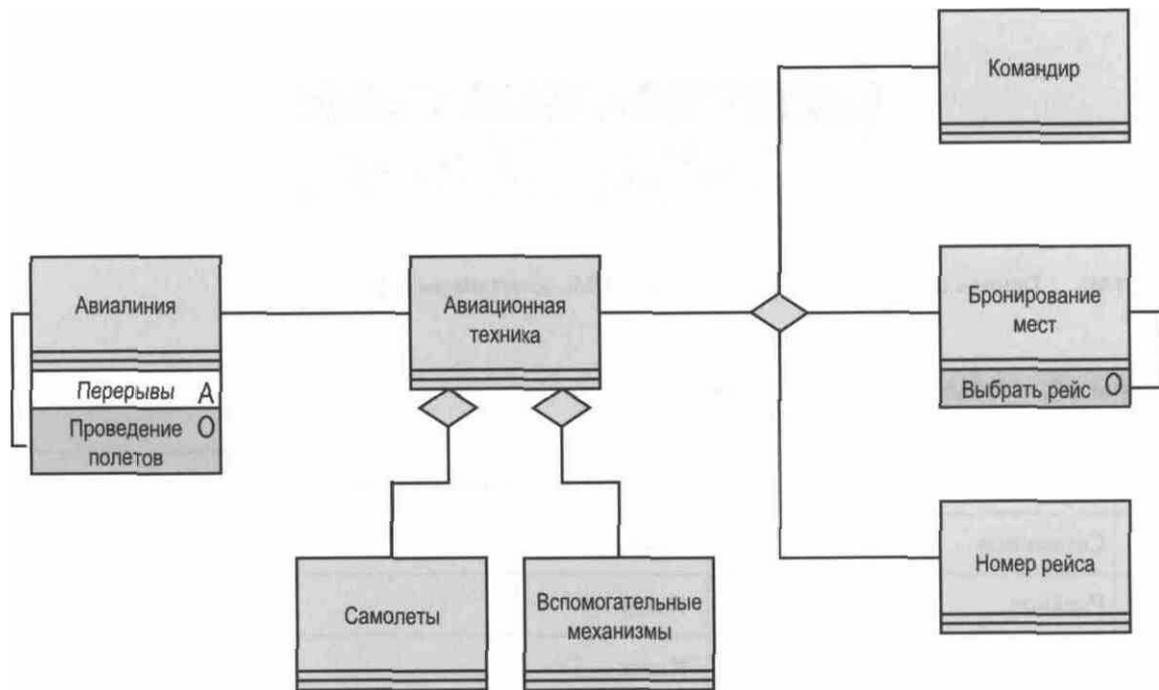


Рис. 147. UML-диаграмма класса: ассоциации

Отношения наследования, которые существуют между классами, представляются связью *обобщает* и изображаются треугольником.

Атрибуты и операции, которые были привязаны к старшему классу, передаются подчиненным классам. Пример таких отношений наследования приведен на рис. 148.

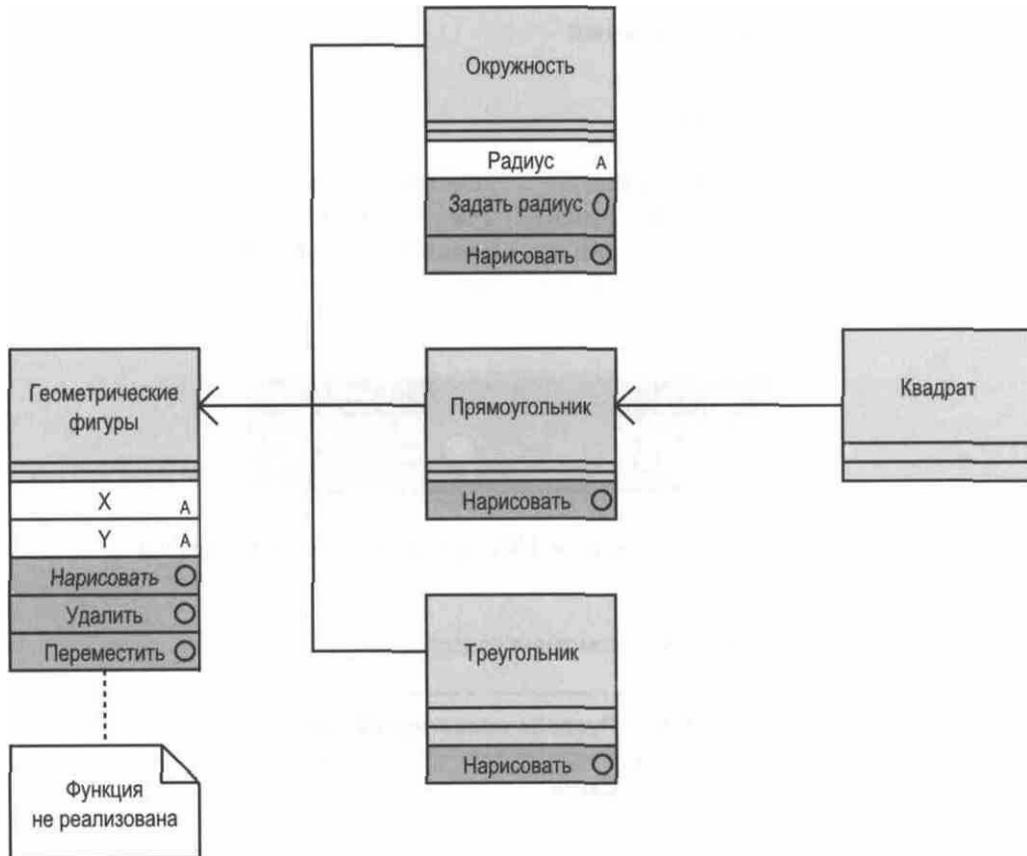


Рис. 148. UML-диаграмма класса: отношения наследования

Дополнительными элементами модели, доступными в UML-диаграмме класса, являются:

- объект *пакет*, используемый для группировки элементов модели;
- объект *примечание*, который предоставляет дополнительную информацию о модели;

- объекты класс и интерфейсы.

*Интерфейс* обозначает интерфейс класса (соединение *поддерживает*). При помощи вызова интерфейса (соединение *вызывает*) другие классы могут использовать объект *класс*, соответствующий объекту *интерфейс*.

### 7.3.10. UML-диаграмма описания класса UML Class description diagram —

UML-диаграмма описания класса при конструировании моделей, показана на рис. 149. Объекты рассматриваемой диаграммы приведены в табл. 38, а допустимые связи — в табл. 39.

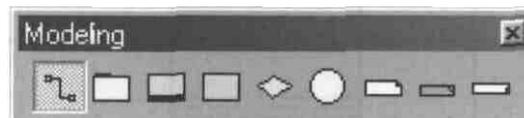


Рис. 149. Панель инструментов для создания UML-диаграммы описания класса

Таблица 38. Объекты UML-диаграммы описания класса

	Английское название объекта	Русское название объекта
	Связь	Связь
	Пакет	Пакет
	Класс	Класс
	Объект	Объект
	Ассоциация	Ассоциация
	Интерфейс	Интерфейс
	Примечание	Примечание
	Операция	Операция
	Атрибуты	Атрибуты

Таблица 39. Связи в UML-диаграмме описания класса

Английское название связи	Русское название связи
Belongs to	Принадлежит к
Has instance	Имеет экземпляр
Has member	Имеет в своем составе
Supports	Поддерживает

### Стандартный и расширенный методологические фильтры. Обзор моделей

UML-диаграмма описания класса — это дополнение к стандартным диаграммам UML, которое позволяет более точно определить классы. Параметры UML-диаграммы описания класса являются подмножеством параметров UML-диаграммы класса. Другими словами, все параметры моделирования в UML-диаграмме описания класса доступны и в UML-диаграмме класса. Объекты *атрибуты*, *операция*, *примечание*, *объект* и *интерфейс* могут быть описаны в UML-диаграмме описания класса.

Это описание может быть сделано также в UML-диаграмме класса, но моделирование с помощью UML-диаграммы описания класса является целесообразным в случае, если UML-диаграмма класса становится слишком перегруженной графически. В этой ситуации UML-диаграмма описания класса должна быть определена как детализация объектов

UML-диаграммы класса. Объекты *атрибуты*, *операция*, *примечание*, *объект* и *интерфейс*, которые могут быть использованы, но не являются необходимыми в UML-диаграмме класса, могут быть перемещены в UML-диаграмму описания класса.

#### 7.3.11. UML-диаграмма взаимодействия — UML Collaboration diagram

Взаимодействия в форме обмена сообщениями между объектами отображаются в UML-диаграмме взаимодействия. Объекты, называемые также экземплярами, — это конкретные экземпляры классов.

Нотация модели приведена на рис. 150, где представлен только один вид связи — *interacts with* (*взаимодействует с*).

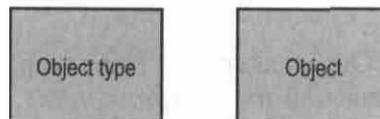


Рис. 150. Изображение объектов в UML-диаграмме взаимодействия: тип объекта и объект

Обмен сообщениями моделируется с помощью соединения *взаимодействует с*. Это соединение указывает конкретные значения атрибутов *условие*, *номер сообщения*, *операция* и *параметр*. Атрибуты определяются следующим образом:

- *Условие (Condition)*. Принимает вид других сообщений, которые должны быть посланы перед посылкой текущего сообщения. Другие сообщения и соответствующие им *номера сообщений* заданы в виде списка. Если не существует никаких предпосланных сообщений, *условие* становится не нужным. Условие отделяется от номера сообщения косой чертой (/).

*Номер сообщения (Message number)*. Это уникальный номер, идентифицирующий сообщение на диаграмме. Сообщения сортируются в порядке возрастания номеров. Если операция, обрабатывающая в данный момент полученное сообщение, посылает несколько дополнительных сообщений, то старый номер дополняется *подномером*. (Пример:

операция получает сообщение 3.4 и посылает два сообщения с номерами 3.4.1 и 3.4.2.) *Номер сообщения* отделяется от операции двоеточием (:).

*Операция (Operation)*. Отображает *операцию* для класса заданного объекта, которая должна быть выполнена

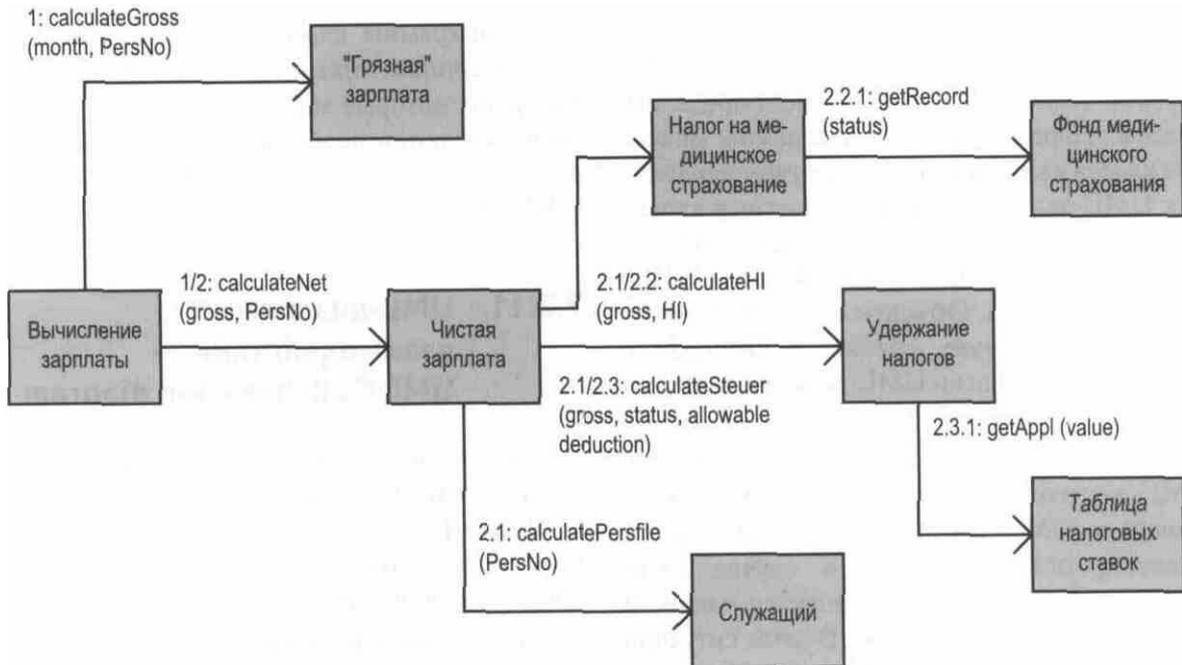


Рис. 151. UML-диаграмма взаимодействия

*Параметр (Parametr).* Определяет список параметров для вызываемой операции. Список параметров заключается в круглые скобки.

Например, 1.3, 2.1/3.2.1: *calculate net (gross, rate)*, где сообщения 1.3 и 2.1 — это условия; сообщение имеет номер 3.2.1; выполняемая операция — расчет чистой зарплаты;

«грязная» зарплата и ставка налога — параметры операции (рис. 151).

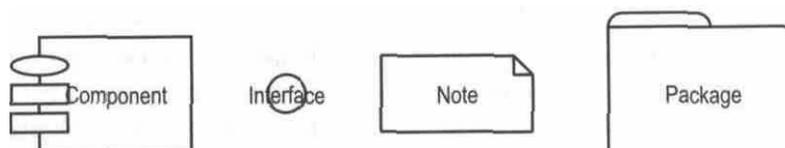


Рис. 152. Изображение объектов в UML-диаграмме компонент: компонента, интерфейс, примечание, пакет

Таблица 40. Связи в UML-диаграмме компонент

Английское название связи	Русское название связи
Belongs to	Принадлежит к
Calls	Вызывает
Contains	Содержит
Depends	Зависит
Supports	Поддерживает

### Component diagram

#### 7.3.12. UML-диаграмма компонент — UML

UML имеет возможность отображать в виде диаграммы такие относящиеся к реализации аспекты, как структура кода (компонента) и структура работы системы. Для этой цели ARIS использует UML-диаграмму компонент. Ее нотация показана на рис. 152. Связи приведены в табл.40.

Компоненты — это элементы, которые формируют блоки в процессе компиляции или компоновки, а также во время выполнения системных операций. Первый тип связей между компонентами описывает физическую структуру компонент. Компонента может содержаться в другой компоненте. Этот факт отображается связью *содержит*.

Второй тип взаимосвязи между компонентами — отношение вызова. Одна компонента вызывает другую через объект *интерфейс*. Компонента, обеспечивающая интерфейс, соединяется с помощью связи *поддерживает* (сплошная стрелка), а компонента, использующая интерфейс, соединяется посредством связи *вызывает* (пунктирная стрелка).

Конфигурация компонент может сохраняться в процессе работы системы. Для этого компоненты группируются и привязываются к объекту *пакет*. Привязка осуществляется с помощью соединения *содержит* между компонентой и ее пакетом. При графическом представлении рекомендуется помещать компоненты в объекты *пакет*.

UML-диаграмма компонент приведена на рис. 153.

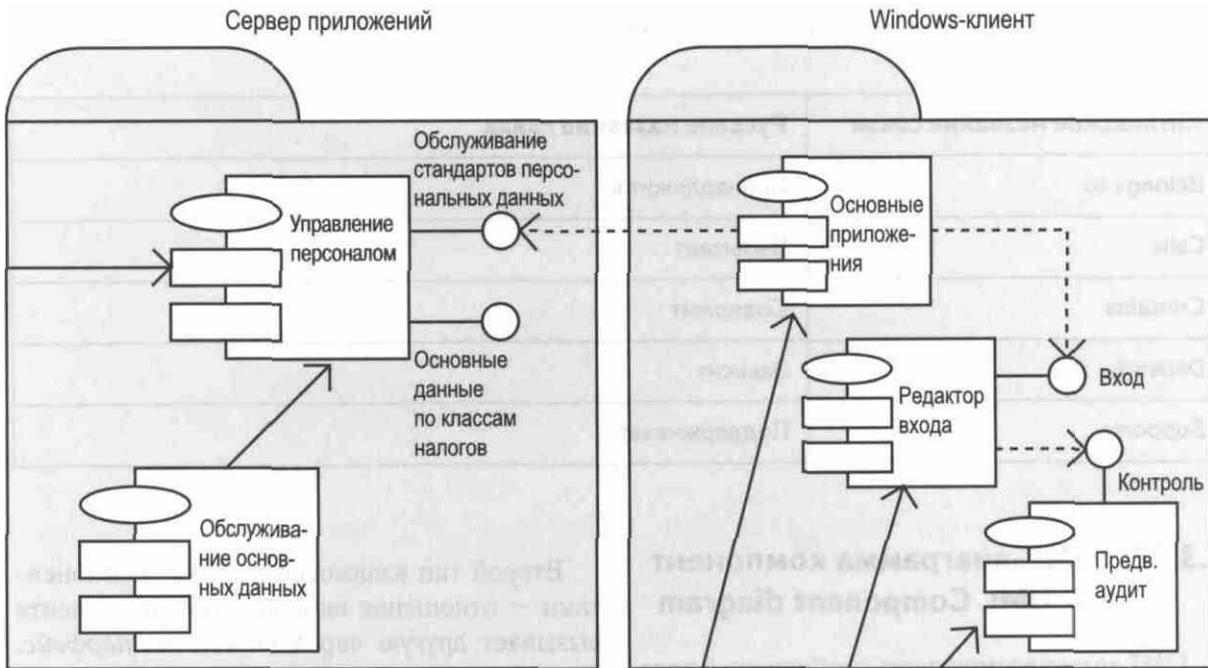


Рис. 153. UML-диаграмма компонент

для представления всех возможных состояний приведена на рис. 154, а возможные связи — в табл.41.

7.3.13. UML-диаграмма состояний — UML State chart diagram

UML-диаграмма состояний предназначена объекта и процесса их смены. Нотация модели



Рис. 154. Изображение объектов в UML-диаграмме состояний: состояние, начальное состояние, конечное состояние, примечание

Таблица 41. Связи объектов в UML-диаграмме состояний

Английское название связи	Русское название связи
Belongs to	Принадлежит
Has transition to	Переходит в

**Стандартный и расширенный методологические фильтры. Обзор моделей**

UML -диаграмма состояний подобно UML-диаграмме действий изображает автоматическое состояние и описывает аналогичную ситуацию. Однако диаграмма состояний сфокусирована на состояниях объектов. Более того, она может содержать действия, относящиеся к состоянию. Действия — предопределенные значения на входе в состояние (вход/), которые выполняются при нахождении в состоянии (выполнить/) или при выходе из состояния (выйти/).

UML-диаграмма состояний ARIS включает объект *состояние*. Переходы между состояниями соединяют состояния с помощью направленных связей (*переходит в*). Так же, как и UML-диаграмма действий, диаграмма состояний должна начинаться с начального состояния и оканчиваться конечным состоянием. На рис. 155 представлена UML-диаграмма состояний.

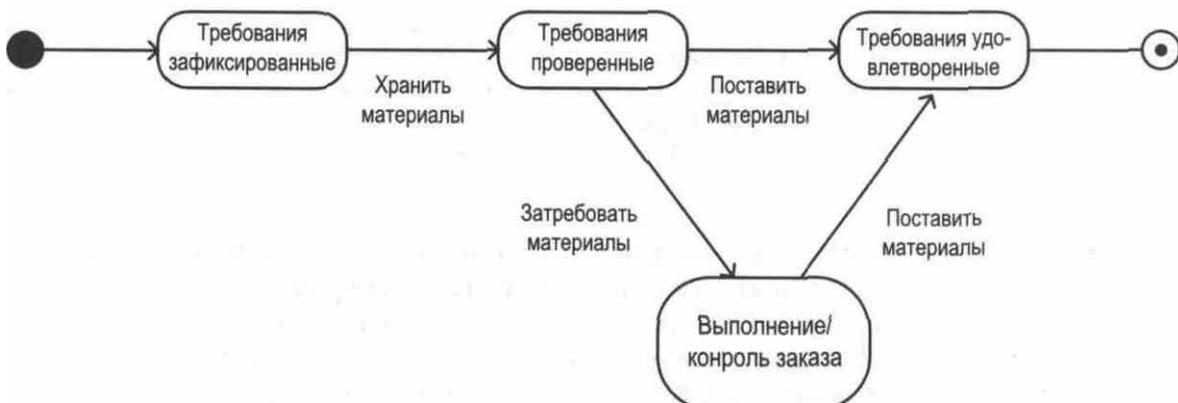


Рис. 155. UML-диаграмма состояний

**7.3.14 UML-диаграмма использования приложений — UML Use case diagram**

Нотация модели приведена на рис. 156 , а возможные связи — в табл. 42.

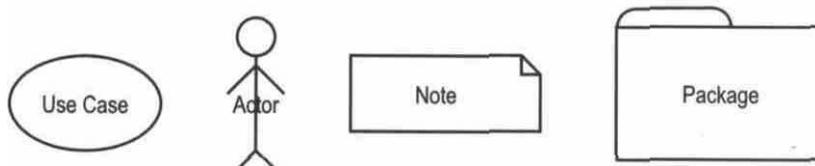


Рис. 156. Изображение объектов в UML-диаграмме использования приложений: пример использования, исполнитель, примечание, пакет

Таблица 42. Связи объектов в UML-диаграмме использования приложений

Английское название связи	Русское название связи
Belongs to	Принадлежит к
Communicates with	Взаимодействует с
Contains	Содержит
Extends	Расширяет
Generalizes	Обобщает
Includes	Включает
Uses	Использует

В UML-диаграмме использования приложений описываются варианты приложений и их исполнители, т.е. привлеченные объекты, на которые воздействуют выбранные приложения. Исполнители изображают пользователей, которые применяют прикладную систему для выполнения своих обязанностей. UML-диаграмма использования приложений описывает внешнее поведение системы с точки зрения пользователя.

Соединения между исполнителями и приложениями определяются как связь *взаимодействует с*. Связь между объектом *пример использования приложения* устанавливается с помощью соединения *обобщает*. В UML-диаграмме доступны также типы объектов *пакет* и *примечание*.

На рис. 157 представлена UML-диаграмма использования приложений.

## Стандартный и расширенный методологические фильтры. Обзор моделей

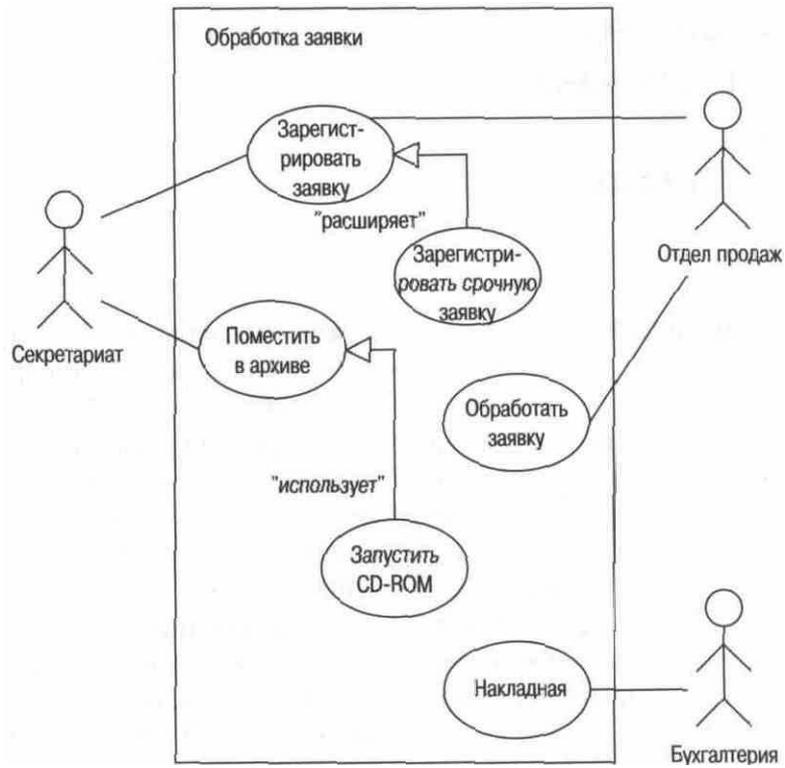


Рис. 157. UML-диаграмма использования приложений

## 8. Обзор некоторых моделей, вошедших в полный методологический фильтр

### 8.1. Дерево продуктов — Product tree

Дерево продуктов предназначено для графического представления и анализа номенклатуры продукции в инфраструктуре организации.

Для формирования матриц выбора процессов необходимо прежде всего сформировать модели входящих и исходящих продуктов (входов-выходов). Под входящими/исходящими продуктами понимаются:

- материальные ресурсы (готовая продукция, реализуемые полуфабрикаты, сырье и материалы);
- финансовые ресурсы (средства платежа: деньги безналичные и наличные, ценные бумаги);

- информационные потоки (счета-фактуры на отгруженную продукцию, заявки потребителей и т.д.).

На данном уровне описания должны рассматриваться не конкретные ресурсы или единицы информационных потоков, а их типы, например, «счет-фактура», «платежное поручение», «прокат» и т.д.

Кроме продуктов, являющихся внешними выходами предприятия, должны быть описаны продукты, являющиеся выходами одних его подразделений и используемые другими его подразделениями (входы).

Нотация модели состоит из обозначения одного объекта — *продукция* — пятиугольником зеленого цвета (рис. 158). Между объектами диаграммы возможна только связь *subsumes* — *содержит*.

Обзор некоторых моделей, вошедших в полный методологический фильтр

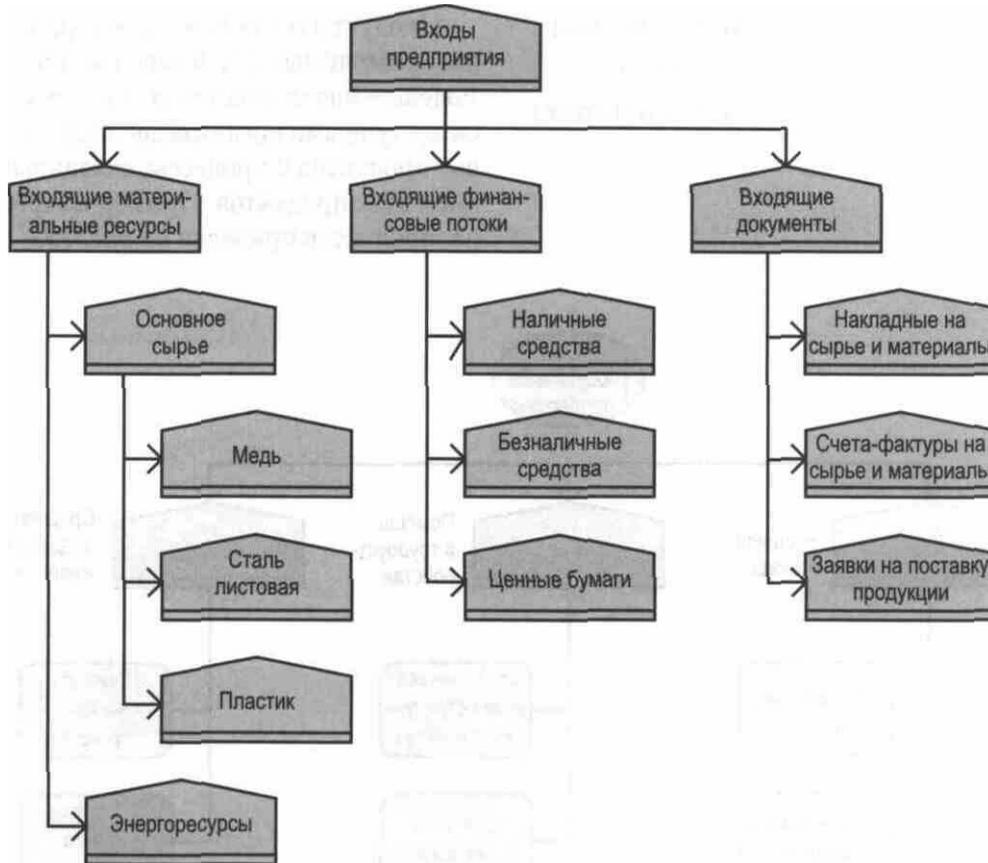


Рис. 158. Дерево продуктов

**8.2. Матрица выбора продукта — Product Selection Matrix**

Матрица выбора продукта предназначена для графического представления

взаимосвязей организационных единиц и выпускаемых ими продуктов. Объекты матрицы выбора продукта представлены на рис. 159.

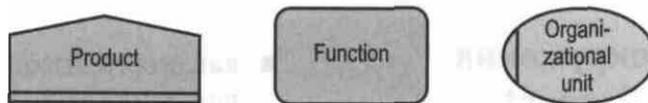


Рис. 159. Объекты матрицы выбора продукта: продукт, функция, организационная единица

В данной модели используются две связи:

- *has output of* — имеет на выходе;
- *is responsible for* — является ответственным за...

Продукты могут быть привязаны к функциям, необходимым для их производства. Данная модель — начальная точка, с которой становятся доступными организационные схемы, деревья продуктов и процессы, связанные с созданием этих продуктов. Пример матрицы выбора продуктов приведен на рис. 160.

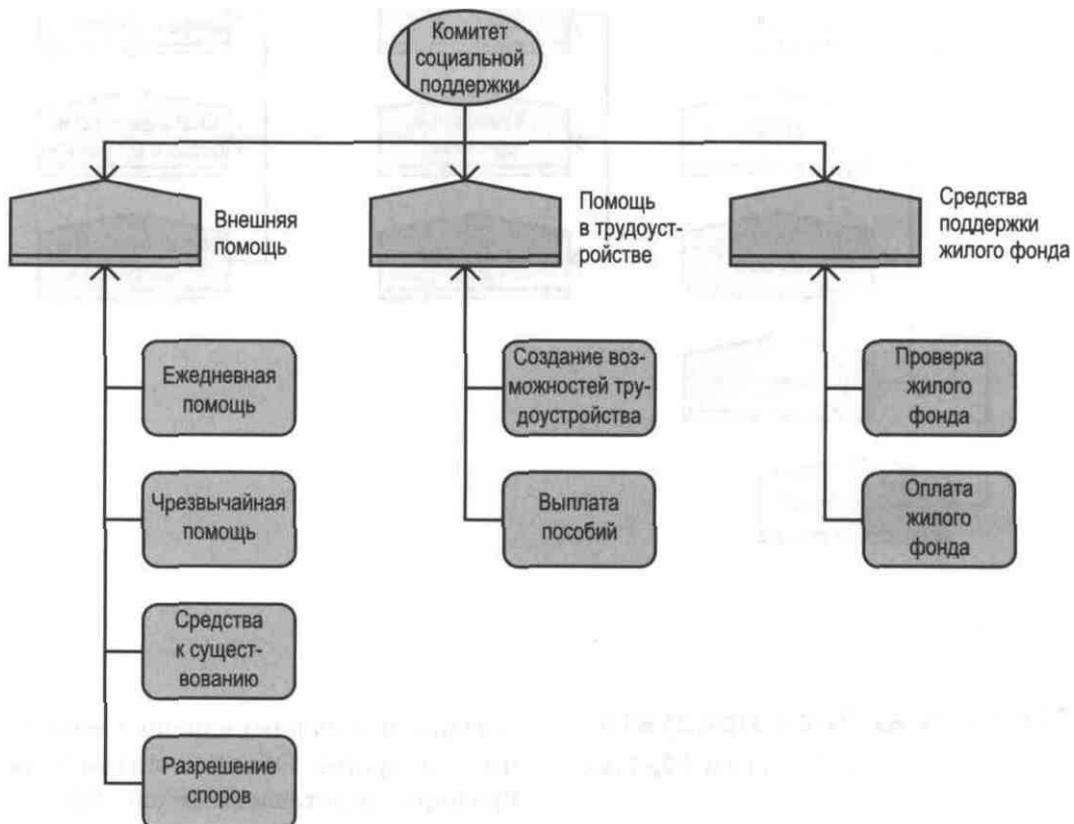


Рис. 160. Матрица выбора продукта для службы социальной поддержки

### 8.3. Диаграмма окружения продукта — Product Allocation Diagram

Диаграмма окружения продукта предназначена для анализа процессов производства продукции. Диаграмма описывает:

- какая организационная единица производит или использует определенные виды продукции;
- какие функции требуются для создания продукции.

Изображение объектов в диаграмме окружения продукта приведено на рис. 161.

Возможные связи между объектами представлены в табл.43.

## Обзор некоторых моделей, вошедших в полный методологический фильтр

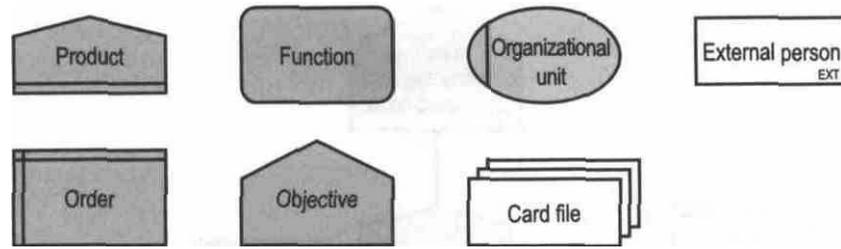


Рис. 161. Объекты в диаграмме окружения продукта: продукт, функция, организационная единица, внештатный сотрудник, нормативный документ, цель, картотека

Таблица 43. Связи объектов в диаграмме окружения продукта

Английское название связи	Русское название связи
Has output of	Имеет на выходе
Help with the creation of	Создается с помощью
Is financially responsible for	Отвечает за финансовые вопросы
Is input for	Является выходом для
Is order basic for	Является основой для упорядочения
Is received from	Направляется к
Is technical responsible for	Отвечает за техническую часть
Is used by	Используется
Provides	Обеспечивает

С помощью диаграммы окружения продукта проводится анализ процесса создания продукта. Аналогично диаграмме обмена продуктов/ услуг этот тип модели может применяться для отображения того, какие организационные единицы предоставляют или используют про-

дукты, какие функции требуются для создания продукта, а также для каких функций продукты являются входом. В диаграмме могут быть описаны и цели, достижению которых способствуют различные продукты. На рис. 162 приведен пример диаграммы окружения продукта.

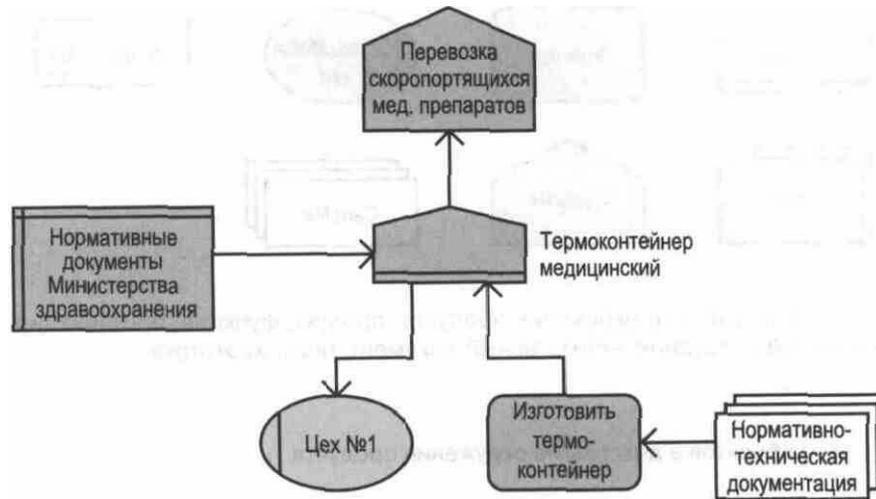


Рис. 162. Диаграмма окружения продукта

### 8.4. Диаграмма взаимодействий — Communication Diagram

Диаграмма взаимодействий позволяет сгруппировать все процессы обмена продук

тами, услугами или информацией согласно их связям с организационными единицами. Изображение объектов в диаграмме приведено на рис. 163.



Рис. 163. Объекты диаграммы взаимодействий: тип организационной единицы, организационная единица, взаимодействие

Модель отражает два вида связей:

- *is received from* — направляется к...;

- *sends* — направляется в ...

Диаграмма взаимодействий представлена на рис. 164.

## Обзор некоторых моделей, вошедших в полный методологический фильтр

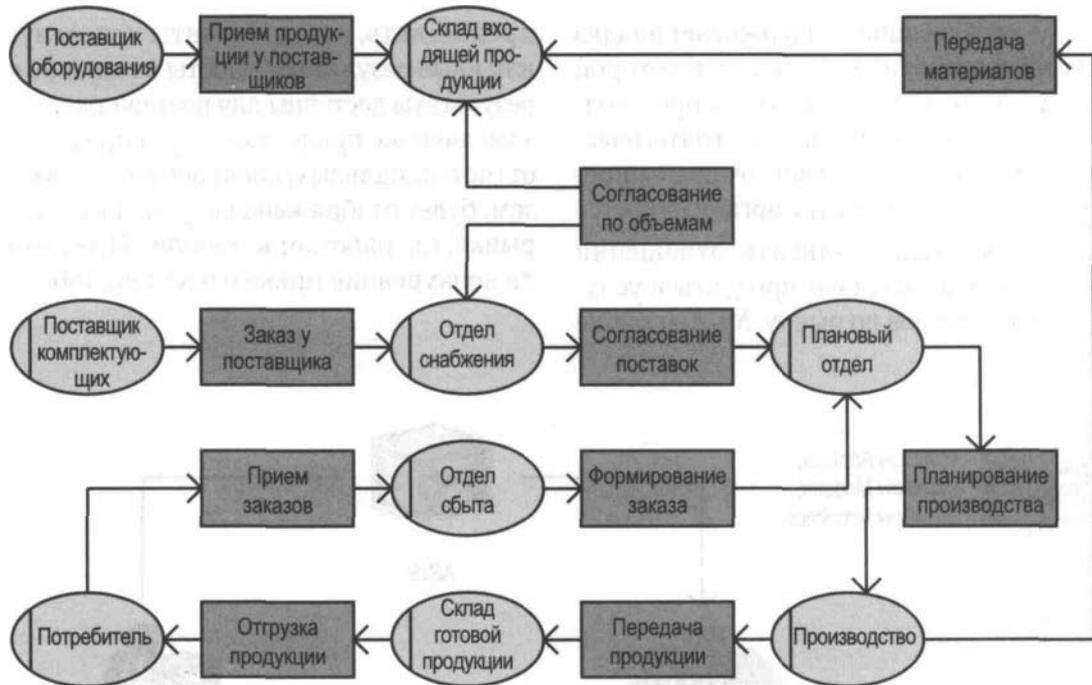


Рис. 164. Диаграмма взаимодействий

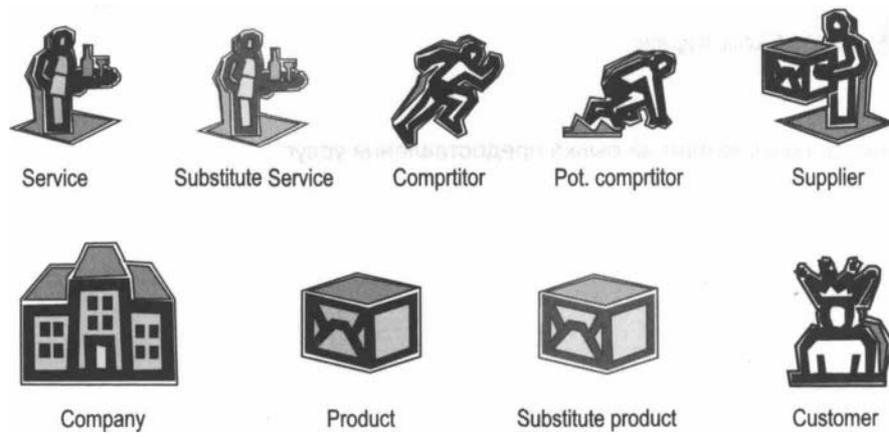
### 8.5. Модель конкуренции — Competition Model

Модель конкуренции предназначена для описания связей, выпускаемой продукции и партнеров. Пиктограммы, используемые в

качестве нотаций модели, показаны на рис. 165.

В модели используются два вида связей:

- *provides* — обеспечивает;
- *is used by* — используется



**Рис. 165. Изображение объектов в модели конкуренции: услуга, замещающая услуга, конкурент, потенциальный конкурент, поставщик, компания, продукт, замещающий продукт, клиент**

Модель конкуренции поддерживает анализ и оценку конкурентной ситуации, в которой находится организация. Структура производства непосредственно влияет на стратегические установки, которые в свою очередь определяют дальнейшее развитие организации.

Модель позволяет описать отношения внутри компании, конечные продукты и услуги, а также партнеров по рынку. Можно также

представить, какие клиенты используют те или иные результаты работы компании, какие результаты доступны для поставщиков и какие заменяющие продукты/услуги предлагаются от (потенциальных) конкурентов. Таким образом, будет отображена ситуация на той части рынка, где работает компания. Пример модели конкуренции приведен на рис. 166.

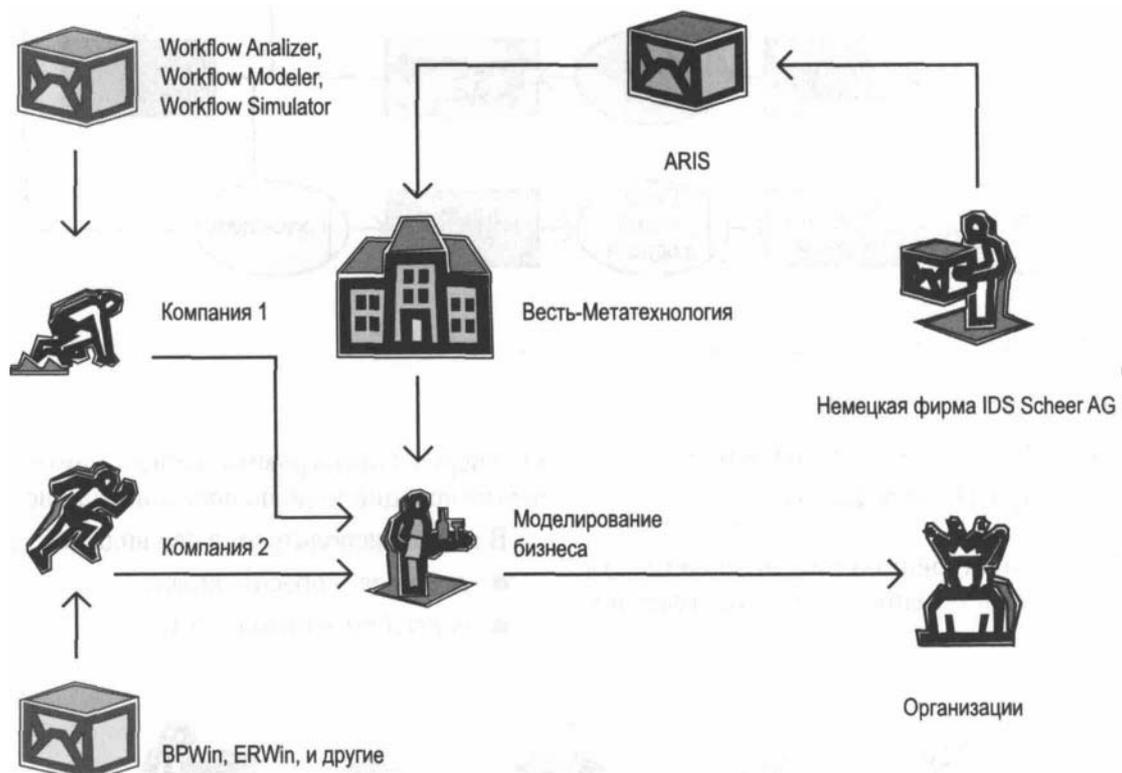


Рис. 166. Модель конкуренции на рынке предоставления услуг

## 8.6. Модель «Технические ресурсы» — Technical resources

При моделировании потоки материалов связываются с отдельными функциями процесса в виде потоков на их входах и выходах. Эти связи описывают преобразование типов материалов, поступающих на вход функции, в типы материалов на выходе функции. Кроме того, это позволяет включить в диаграммы процессов информацию о технических ресурсах, которые необходимы для преобразования материалов.

Для описания используемых технических ресурсов в ARIS введена модель *Технические ресурсы*. Она относится к организационным моделям уровня описания реализации. При помощи модели можно иерархически упорядочить ресурсы, присвоить им тип и классифицировать.

Объекты, входящие в модель *Технические ресурсы*, можно условно разбить на две группы. В первую входят некоторые объекты организационной схемы (см. раздел 6.1.1), а именно:

- организационная единица;
- системная организационная единица;
- тип системной организационной единицы;
- группа;
- местоположение;
- должность;
- штатный и внештатный сотрудники.

Эти объекты необходимы для привязки элементов организационной структуры к имеющимся техническим ресурсам.

Во вторую группу входят собственно технические ресурсы:

- операционные ресурсы (operating resource);
- складское оборудование (warehouse equipment);
- транспортные системы (transport systems);
- вспомогательные технические средства (technical operating supply), их типы и классы (рис. 167).

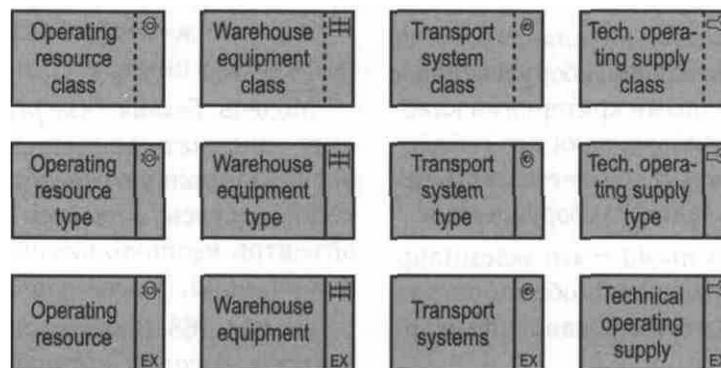


Рис. 167. Объекты, описывающие технические ресурсы

### Моделирование бизнеса. Методология ARIS

*Операционные ресурсы* — это экземпляры различных типов операционных ресурсов, которые доступны для выполнения задач, стоящих перед организацией. Операционные ресурсы часто идентифицируются с помощью различных инвентарных номеров (например, номер завода).

*Тип операционных ресурсов* представляет собой обобщение совокупности различных операционных ресурсов, которые имеют одинаковую технологическую базу.

*Классы операционных ресурсов* являются обобщением типов операционных ресурсов в соответствии с различными критериями классификации. Следовательно, один тип операционных ресурсов может соответствовать нескольким классам операционных ресурсов.

*Складское оборудование* — это экземпляры различных типов складского оборудования, которые доступны и предназначены для выполнения стоящих перед организацией задач. Складское оборудование часто идентифицируется присвоенными ему номерами складов.

*Тип складского оборудования* представляет совокупность нескольких видов складского оборудования, которые имеют одинаковую технологическую базу.

*Класс складского оборудования* является обобщением типов складского оборудования в соответствии с различными критериями классификации. Следовательно, один тип складского оборудования может соответствовать нескольким классам складского оборудования.

*Транспортная система* — это экземпляр типа транспортной системы. В общем случае она может быть идентифицирована инвентарным номером.

*Тип транспортной системы* представляет обобщение совокупности нескольких транспортных систем, имеющих одинаковую технологическую базу.

*Класс транспортной системы* является обобщением типов транспортных систем в соответствии с различными критериями классификации. Следовательно, один тип транспортной системы может соответствовать нескольким классам транспортных систем.

*Вспомогательные технические средства* — это экземпляр типа вспомогательных технических средств. В общем случае он может быть идентифицирован посредством инвентарного номера.

*Тип вспомогательного технического средства* представляет обобщение совокупности нескольких вспомогательных технических средств, имеющих одинаковую технологическую базу.

*Класс вспомогательных технических средств* является обобщением их типов в соответствии с различными критериями классификации. Один тип вспомогательных технических средств может соответствовать нескольким классам вспомогательных технических средств.

Возможности иерархического построения моделей *Технические ресурсы* позволяют описывать структуру технически сложных производственных объектов (заводов, фабрик и т.д.), а также отображать их компоненты и связи между ними.

Модель *Технические ресурсы* также позволяет описывать размещение рабочих мест и организационную ответственность за технические ресурсы. Для этого используются типы объектов *местоположение, организационная единица, должность и сотрудник*.

На рис. 168 приведен пример модели *Технические ресурсы*, которая описывает оборудование, используемое для контроля качества продукции, а также показаны возможные связи между объектами модели.

## Обзор некоторых моделей, вошедших в полный методологический фильтр

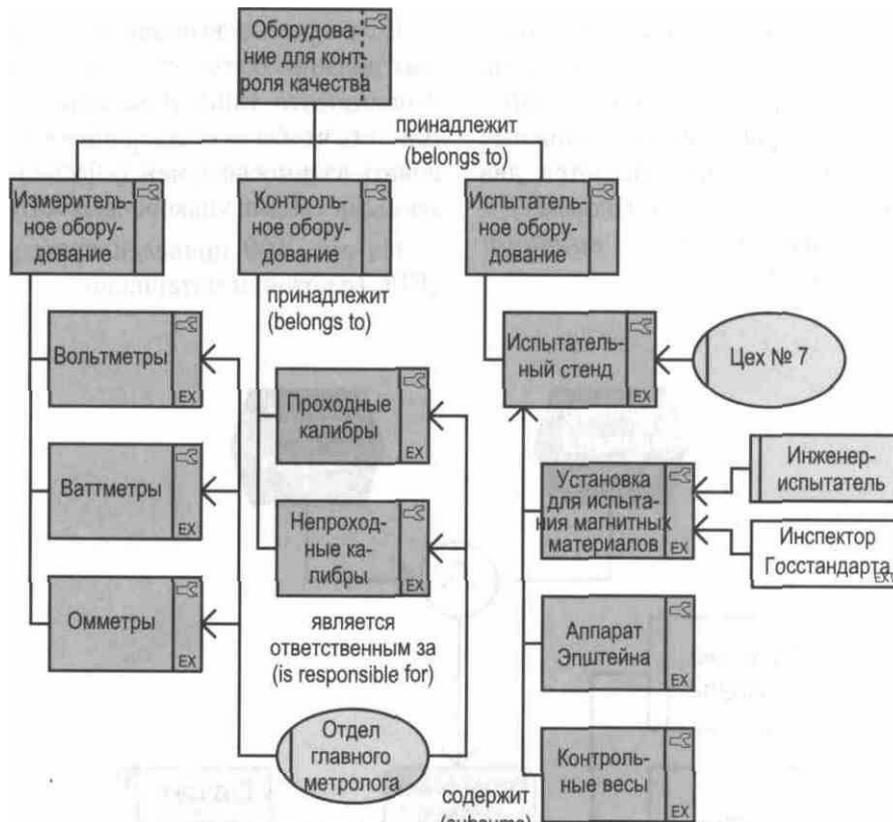


Рис. 168. Модель «Технические ресурсы»

## 8.7 Модели, предназначенные для описания материалов

### 8.7.1. Диаграммы eEPC (с потоком материалов) — Extended event driven process chain (material flow)

В методологии ARIS поток материалов в бизнес-процессах может представляться с помощью диаграммы eEPC (с потоком материалов), которая является расширением обсуждавшейся ранее диаграммы eEPC (см. раздел 6.4.1). Кроме объектов, отражаемых в диаграмме eEPC, в рассматриваемой диаграмме используются также следующие типы объектов:

- тип материала (material type) (рис. 170);

- тип упаковочного материала (packing material type) (рис. 170);
- операционные ресурсы (operating resource), их типы и классы (рис. 167);
- складское оборудование (warehouse equipment), его типы и классы (рис. 167);
- транспортные системы (transport systems), их типы и классы (рис. 167);
- вспомогательные технические средства (technical operating supply), их типы и классы (рис. 167).

Объект *тип материала* может быть связан с объектом *функция* посредством входящего или выходящего соединения. Соответственно объект *тип материала* будет описывать входные (перерабатываемые объектом *функция*) и выходные (создаваемые объектом *функция*) материалы

### Моделирование бизнеса. Методология ARIS

Для преобразования материалов требуются технические ресурсы. В цепочке процессов их можно связать с объектами типа *функция*. Для того чтобы специфицировать возможные альтернативные ресурсы, используются два типа соединений — *требуется* (*requires*) и *требуется альтернативный вариант* (*requires alternatively*).

Если в рамках выполнения функции материалы должны быть упакованы, то необходимо использовать типы упаковочных материалов. Для того чтобы их специфицировать, надо установить взаимосвязь между функцией и используемыми типами упаковочных материалов.

На рис. 169 приведен пример диаграммы eEPC (с потоком материалов).

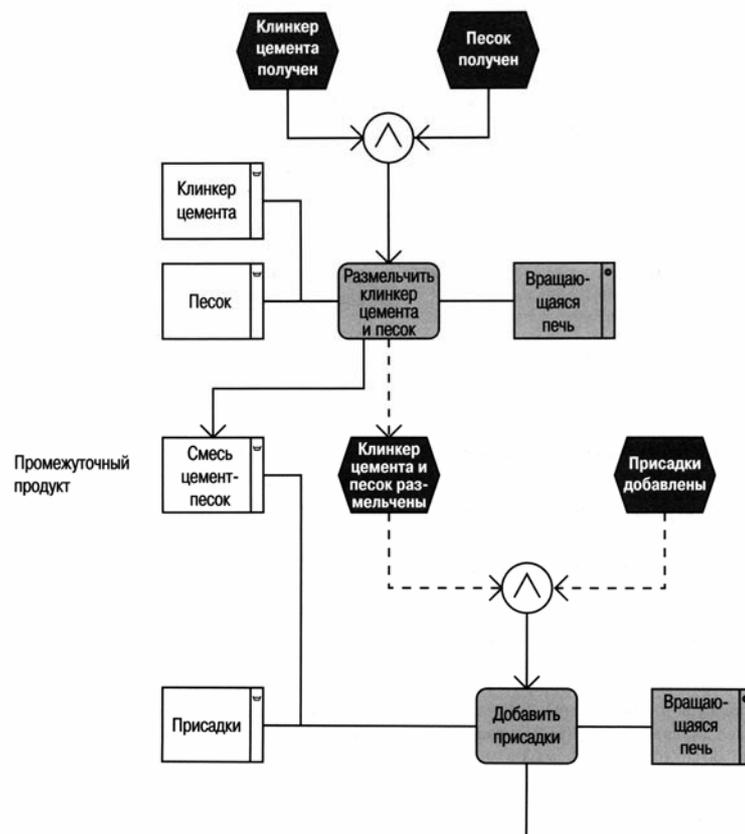


Рис. 169. Диаграмма eEPC (с потоком материалов)

### 8.7.2. Диаграмма материалов — Material Diagram

Диаграмма материалов предназначена для описания типов материалов, их

упорядочивания в соответствии с иерархией и классификации по классам материалов. Она относится к моделям данных уровня определения требований. Объекты этой диаграммы показаны на рис. 170.

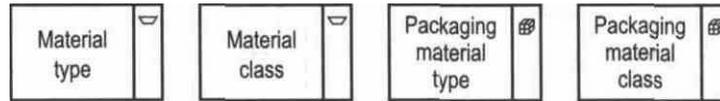


Рис. 170. Объекты в диаграмме материалов: тип и класс материала, тип и класс упаковочного материала

*Тип материала* — это совокупность материалов, имеющих одинаковые характеристики. Типы материалов могут объединяться, образуя класс материалов. При таком объединении проблема схожести (т.е. отнесение материала к некоторому классу) решается с помощью различных критериев классификации. Другими словами, один тип материала может соответствовать нескольким классам материалов.

Типы материалов могут быть связаны с типами упаковочных материалов. Это означает, что определенные типы материалов могут перемещаться только в том случае, если для их упаковки используется конкретный тип упаковочных материалов.

*Тип упаковочного материала* — это совокупность отдельных упаковочных материалов, имеющих одинаковые характеристики. Упаковочный материал также может быть определен, классифицирован и иерархически упорядочен. Это позволяет, например, описать структуру и ограничения сложных упаковочных товарных комплексов.

Типы материалов могут быть объединены в классы упаковочных материалов. При таком объединении используются различные критерии классификации. Иначе говоря, один тип упаковочного материала может соответствовать нескольким классам материалов.

На рис. 171 приведена диаграмма материалов, включающая иерархию уровней и классификацию



Рис. 171. Диаграмма материалов

### 8.7.3. Диаграмма материальных потоков — Material Flow Diagram

Диаграмма материальных потоков отображает поток материалов между функциями. Она относится к моделям процессов/управления уровня определения требований. Разработка данной модели аналогична построению диаграммы информационных потоков (см. раздел 7.3.1). В

диаграмме материальных потоков две функции связываются посредством объекта, представляющего поток материалов. Такое соединение указывает на перемещение потока материалов от исходной функции к целевой. Если необходимо специфицировать этот поток более подробно, то его можно детализировать другой диаграммой материальных потоков, выстраивая тем самым иерархию их описаний.

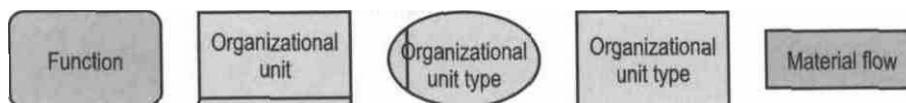


Рис. 172. Объекты диаграммы материальных потоков: функция, организационная единица, тип организационной единицы (2 объекта), материальный поток

Обзор некоторых моделей, вошедших в полный методологический фильтр

и управление ими. Она относится к моделям процессов/управления уровня определения требований.

## 8.8. Диаграмма управления бизнесом — Business Control Diagram

Диаграмма управления бизнесом описывает все потенциальные риски бизнес-процесса



Рис. 173. Объекты диаграммы управления бизнесом: документ, организационная единица, риск, решение, управляющее воздействие

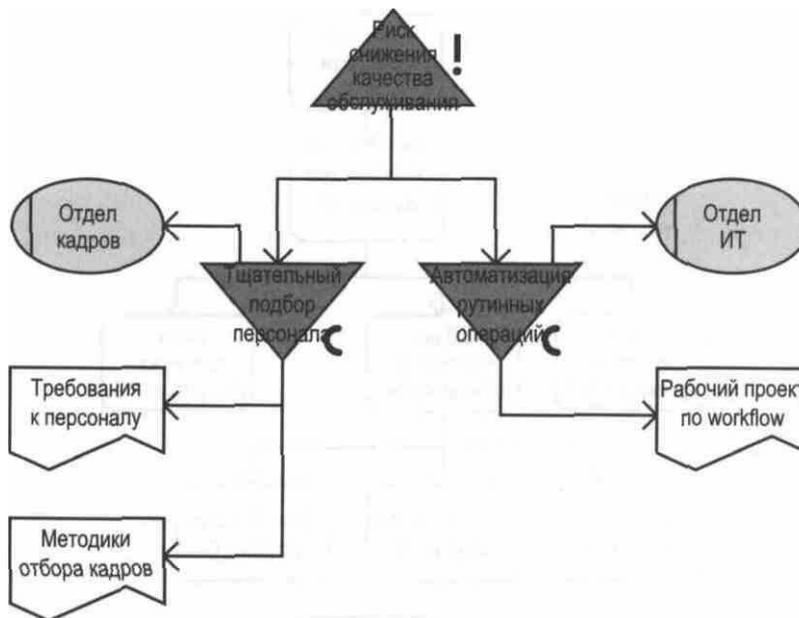


Рис. 174. Диаграмма управления бизнесом

*Риск* означает потенциальную опасность для процесса не достигнуть желаемой цели. Управление риском — это путь исключения риска или уменьшения его степени. Решение по риску означает реализацию управления отдельным риском.

*Диаграмма управления бизнесом* по структуре аналогична матрице или таблице. Абсцисса соответствует потенциальным рискам

процесса, а ордината показывает потенциально возможные способы управления рисками. *Объекты решения по рискам* в качестве операторов размещаются между объектами *риск* и *управление риском*. В модель дополнительно могут быть включены организационные единицы и документы, которые поддерживают реализацию управления соответствующим риском. Пример диаграммы приведен на рис. 174.

## 8.9. Модель структуры — Structuring Model

*Модель структуры* относится к моделям процессов/управления уровня определения требований и предназначена для представления иерархии или другого системного упорядочивания фактов, их обобщения или детализации.

Диаграмма содержит только один объект — структурный элемент, который

представляет некоторый факт (в соответствии с предполагаемой систематизацией). Тип связи показан на рис. 175.

Модели, имеющие отношение к фактам, могут быть «привязаны» к структурным элементам, составляющим иерархию фактов. Модели структуры применяются чаще всего в менеджменте качества. На рис. 175 показана иерархия документов системы менеджмента качества (СМК), представленная в виде модели структуры

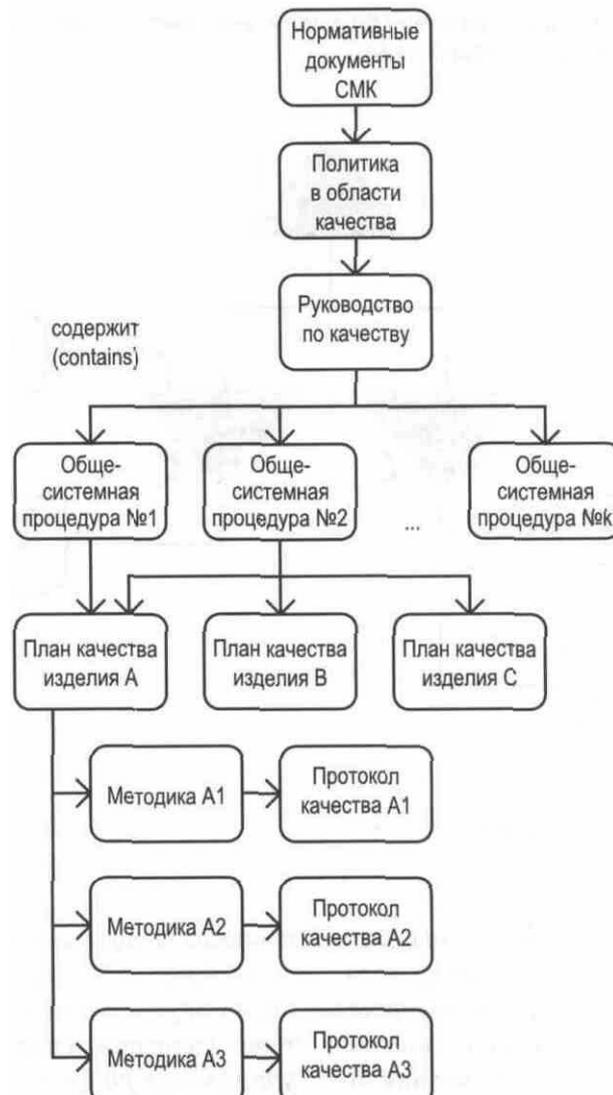


Рис. 175. Нормативные документы системы менеджмента качества, представленные в модели структуры

## 8.10. Диаграмма ролей — Role Diagram

Диаграмма ролей предназначена для более точного описания процессов. Ее главным назначением является описание прав, обязанностей и ролей организационных единиц, участвующих в процессах. Она относится к моделям процессов/управления уровня определения требований.

Диаграмма ролей содержит объекты рассмотренных ранее организационной схемы — организационная единица и ее тип; группа; штатный и внештатный сотрудники; тип служащего; должность; местоположение (раздел 6.1.1), модели технических терминов — технический термин (раздел 6.3.1), карты знаний — категория знания (раздел 7.3.4) и дерева функций — функция (раздел 6.2.1).

Новым объектом является объект *полномочия* {Authorization condition} (на рис. 176

— это объект *разрешение директора*). Объект описывает условия, при которых могут выполняться действия, разрешенные для организационных элементов, т.е. их роли. Роль — это участие организационной единицы или служащего в процессе при определенных условиях, необходимых для выполнения процесса или его части.

Роль может быть задана для служащих, должностей или информационных систем. Она формирует взаимосвязь между процессами и ресурсами, вовлеченными в них.

Связи в диаграмме ролей показаны на рис 176



Рис. 176. Диаграмма ролей

## 9. Общие принципы моделирования в ARIS

Моделирование деятельности подразумевает создание модели, адекватно отражающей реальный объект — организацию. При этом моделирование должно проходить в соответствии с соблюдением ряда требований и выполнением набора правил, однозначных для понимания и единых для всех участников моделирования.

К подобным правилам относятся:

- выделение набора используемых моделей, интерпретация выбранных моделей, принятие правил их построения;
- структурирование информации при создании модели организации — заданные правила размещения моделей и объектов в иерархии папок базы данных, а также определение прав доступа участников к разрабатываемым моделям;
- определение глубины моделирования;
- задание правил именования объектов и моделей;
- определение правил семантической проверки созданных моделей.

### 9.1. Выбор типов моделей

Интегрированная система ARIS содержит множество моделей, каждая из которых отнесена к определенному типу представления и уровню описания. Полный набор моделей является, безусловно, избыточным для проведения конкретного проекта, так как существуют группы моделей, представляющих альтернативные способы описания.

Кроме того, использование большого количества различных типов моделей и нотаций почти всегда означает дублирование описания, то есть представление одной и той же информации с нескольких точек зрения и на разных уровнях обобщения. Нельзя сказать, что подобное описание является некорректным или неоправданным. Однако при этом следует помнить о том, что резко возрастают трудозатраты на поддержание моделей в актуальном состоянии, а также увеличивается вероятность занесения противоречивой информации, наличие

которой зачастую слишком сложно отследить и исправить.

При подготовке каждого проекта, проводимого в среде ARIS, необходимо определиться с набором используемых типов моделей, а следовательно, объектов, их атрибутов и связей.

Выбор методов моделирования зависит от описываемого объекта и перечня задач, которые должны быть решены. Например, для проекта по внедрению системы управления предприятием SAP R/3 необходимо применение специальных методов моделирования, отвечающих требованиям отображения референтных моделей этой системы. К ним относятся модель структуры процессов (для выбора вариантов сценариев, процессов, функций), модель данных SAP SERM, а также набор объектов, позволяющих отразить отдельные сущности системы — системные организационные единицы (для отражения внутренней организационной структуры системы), экранные интерфейсы (для отображения транзакций системы) и многие другие.

При выборе моделей, которые будут использоваться для комплексного описания системы управления и ее окружения, необходимо учитывать следующие обстоятельства:

- каждая модель или группа взаимосвязанных моделей представляют собой различные методы моделирования, отражающие специфические взгляды на правила представлений элементов и явлений предметной области;
- две или несколько моделей могут дополнять друг друга. Например, набор моделей eEPC отражает логику выполнения процессов и функций на разных уровнях обобщения. Модель *дерево функций* представляет иерархию процессов и функций, входящих в модели eEPC на разных уровнях, никак не показывая их последовательность и условия выполнения;

- две или более модели могут быть альтернативой друг другу. Широко применяемая модель eERM и объектная модель ОМТ являются в некотором роде альтернативными, представляя различные взгляды на правила моделирования данных. Хотя надо отметить, что объектно-ориентированное моделирование гораздо шире подхода «сущность-отношение», и возможно их совместное применение в рамках одной модели организации;

- в каждом проекте должен быть использован свой, ограниченный и настроенный набор моделей, основанный на методологии ARIS;

- задание методологии моделирования для каждого конкретного случая осуществляется путем настройки методологических фильтров ARIS и назначения этих фильтров соответствующим пользовате

лям и их группам, участвующим в проводимых работах;

- методологические фильтры ограничивают методологию ARIS до объема, требуемого для создания модели организации. Кроме того, необходимо оговорить правила и осуществить настройку графических элементов — форм, размеров и цветовой гаммы объектов, относительное расположение объектов на модели, вынесение характеристик объектов на графику моделей;

- выбор моделей напрямую зависит от целей моделирования и ожидаемого результата. В этом случае трудно дать строгое и однозначное соответствие моделей решаемым задачам. Однако все-таки можно предложить некоторые рекомендации по выбору моделей. Часть из них приведена в табл.44.

Таблица 44. Рекомендации по выбору моделей ARIS

Задачи моделирования	Используемые модели
Общее описание организации: - определение целей; - описание организационной структуры; - выделение и описание процессов верхнего уровня; - описание функций и процессов; - документирование деятельности.	Диаграмма целей. Организационная схема. Диаграмма цепочек добавленного качества (VAD). Дерево функций. eEPC (событийная цепочка процесса). Диаграмма цепочки процесса (PCD). Диаграмма окружения функции. Офисный и производственный процессы.
Разработка программного обеспечения: - проектирование структуры программной системы; - подготовка проектной документации; - проектирование системного интерфейса; - проектирование интерфейсов пользователей; - структурирование данных; - проектирование базы данных; - администрирование баз данных.	Организационная схема. Расширенная eERM модель "сущность-отношение". eEPC (событийная цепочка процесса). Диаграмма прикладной системы. Блок-схема программы. Модель экранного дизайна. Модель "Топология сети". Диаграмма сети. Модель данных IEF. Диаграмма отношений. Диаграмма классов.
Функционально-стоимостной анализ	Организационная схема. Диаграмма стоимостных драйверов. Диаграмма категории затрат. Разновидности модели eEPC. Диаграмма цепочки процесса (PCD). Модель PCD (с потоком материалов). Диаграмма цепочек добавленного качества (VAD).
Внедрение системы управления SAP R/3	Организационная схема. Дерево функций. Структурированная модель SAP SERM "сущность-отношение". Диаграмма взаимодействий. Разновидности модели eEPC. Диаграмма окружения функций.
Управление знаниями	Карта знаний. Диаграмма структуры знаний. Разновидности модели eEPC.

## 9.2. Структурирование информации при моделировании

Проведение комплексного проекта, подразумевающего создание большого количества взаимосвязанных моделей, требует продуманного подхода к структуризации информации. Необходимо разработать такую структуру модели организации, которая бы позволила осуществлять быстрый поиск требующейся

информации, обеспечивать контролируемый доступ участников с распределением ответственности.

В качестве механизма формирования подобной структуры в среде ARIS существует возможность разработки (см. раздел 4.1) иерархии папок — аналога файловой структуры. В этом смысле папки выступают местом хранения объектов и моделей (рис. 177).

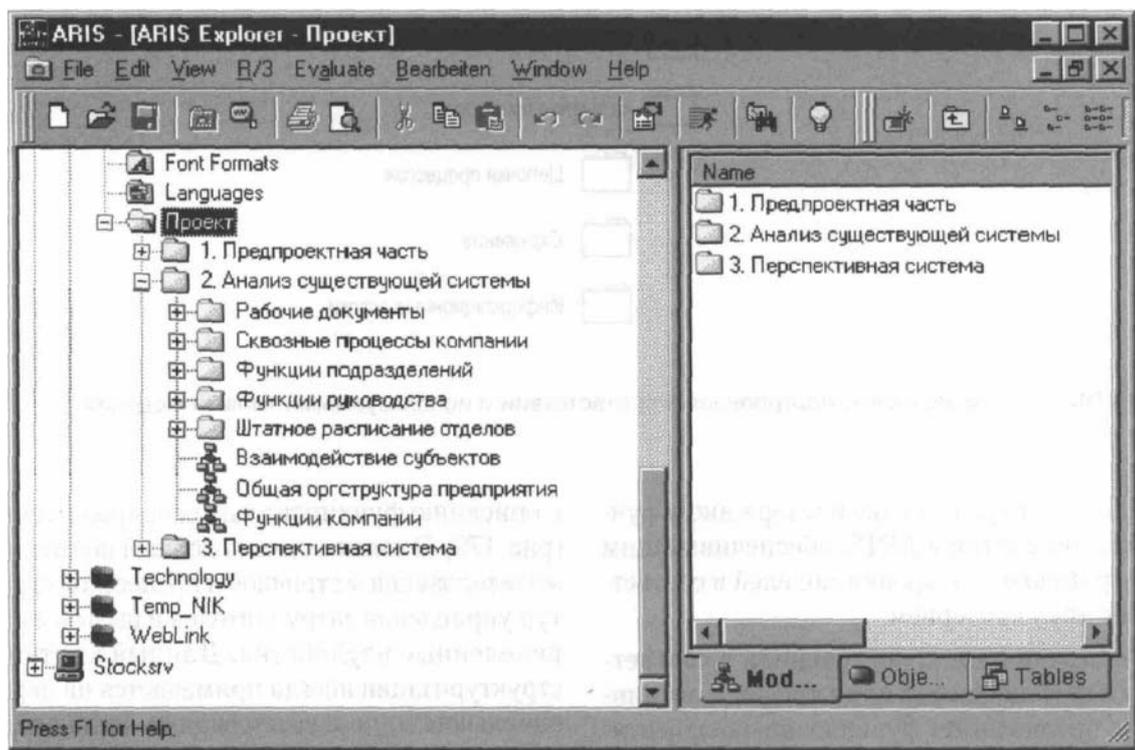


Рис. 177. Структурирование информации при помощи папок

При создании иерархии папок может быть использовано несколько критериев структуризации, основанных на:

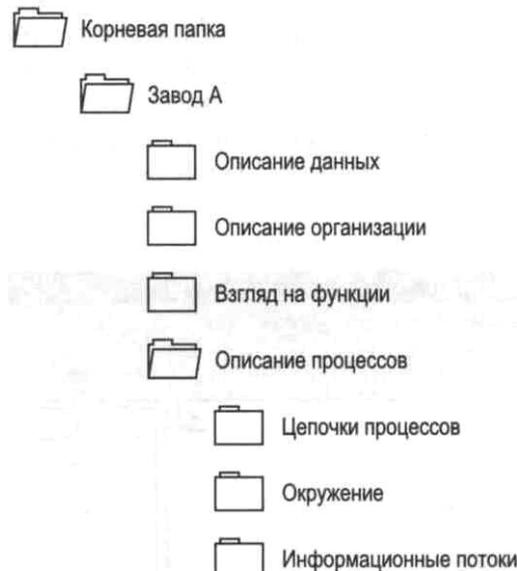
- типах используемых моделей;
- функциональной структуре компании;

структуре процессов компании; этапах проведения работ по моделированию; модулях и функциях стандартных систем управления; комбинации указанных выше критериев.

### Моделирование бизнеса. Методология ARIS

Иерархия папок, построенная в соответствии с типами моделей (рис. 178), используется редко, так как концепция ARIS сама по себе подразумевает разделение типов

предметных областей на типы представления и уровни описания, внутри которых выделяются

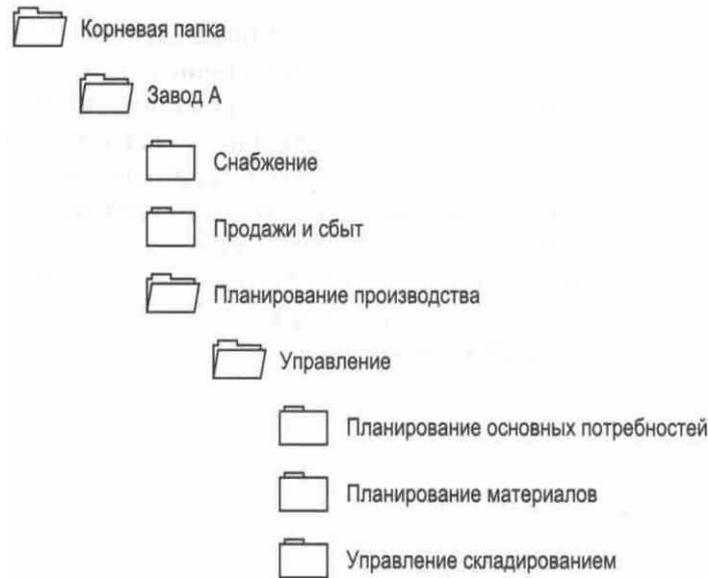


**Рис. 178. Иерархия папок, построенная в соответствии с используемыми типами моделей**

моделей. И это разделение поддержано и функционалом системы ARIS, обеспечивающим фильтрацию и сортировку моделей в соответствии с этим критерием.

Иерархия папок, построенная в соответствии с функциональным разбиением организации, предполагает функционально-ориентированное моделирование и подразумевает группировку моделей по их принадлежности к описанию функциональных

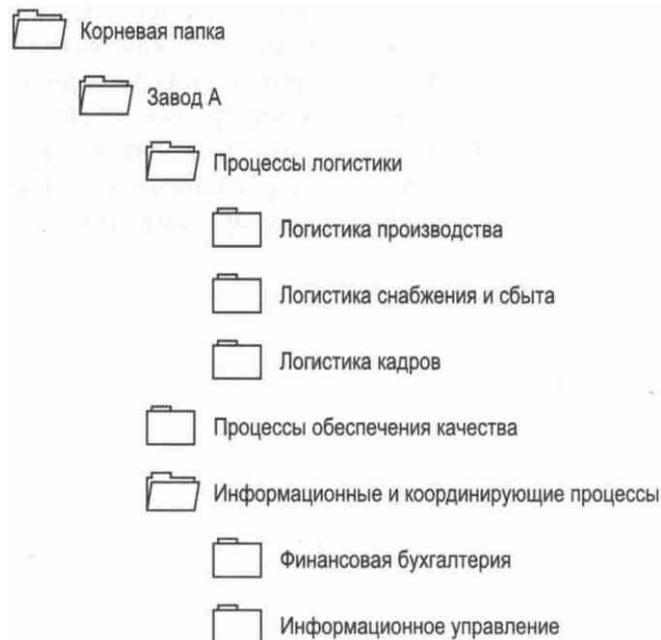
подразделений (рис. 179). В связи с этим подобный подход при моделировании матричной и процессной структур управления затруднителен и вызывает определенные неудобства. Данный критерий структуризации иногда применяется на первоначальном этапе моделирования, когда анализируемая организация рассматривается с точки зрения задач отдельных подразделений и взаимодействия между ними.



**Рис. 179. Иерархия папок, построенная в соответствии с функциональным разбиением организации**

При использовании иерархии папок, построенной в соответствии со структурой процессов, упор делается на сквозные процессы организации, охватывающие все функциональные подразделения (рис. 180). Таким образом, становится возможным применение процессно-ориентированного

подхода к созданию моделей организации и их систематизации. Все стандартные отраслевые модели, поставляемые совместно с интегрированной средой ARIS, построены по этому принципу. Иерархию групп, отражающую структуру процессов, следует считать основной

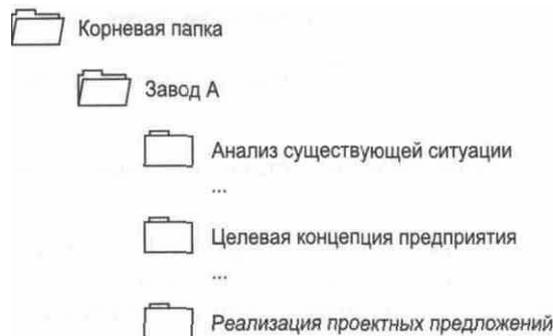


**Рис. 180. Иерархия папок, построенная в соответствии со структурой процессов**

### Моделирование бизнеса. Методология ARIS

Для организации, работающей в сфере услуг, целесообразно расширение данного подхода путем формирования моделей сквозных процессов предоставления отдельных услуг или их групп. Это позволяет оценивать деятельность организации не только с точки зрения процессов, но и в аспекте предоставления услуг. Таким образом, возможна оптимизация процессов, направленная на повышение конкурентоспособности на рынке и управление издержками в разрезе услуг.

Один из важных критериев структуризации — выделение папок в соответствии с этапами моделирования (рис. 181). Этот критерий целесообразно использовать на самом верхнем уровне иерархии, выделяя тем самым основные этапы (подготовка, анализ существующей ситуации — модели «как есть», создание целевой концепции — модели «как должно быть» и т.д.).



**Рис. 181. Иерархия папок, построенная в соответствии с этапами моделирования**

При внедрении комплексных систем управления (например, ERP) необходимо создать такую структуру, которая бы позволяла выстраивать процессы в соответствии с функциональными особенностями системы, обеспечивая удобное средство отображения возможностей системы

и ее планируемых расширений (рис. 182). Однако такая методика не всегда удобна при процессно-ориентированном подходе к внедрению систем, так как она предполагает акцент на структуризации в аспекте устройства самой системы, а не процессов организации.

## Общие принципы моделирования в ARIS

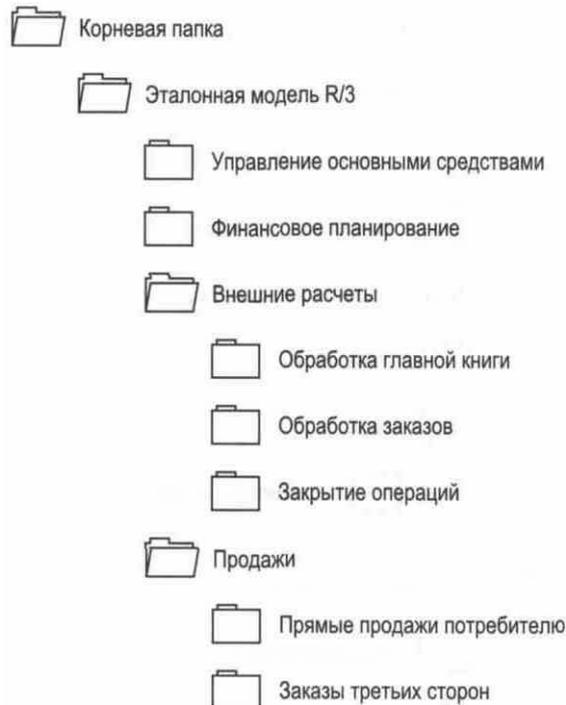


Рис. 182. Иерархия папок, построенная в соответствии с функциональными особенностями системы управления

В любом комплексном проекте используется не один критерий, а их комбинация. Возможные варианты совмещения критериев при построении иерархии папок представлены на рис. 183 — 185.

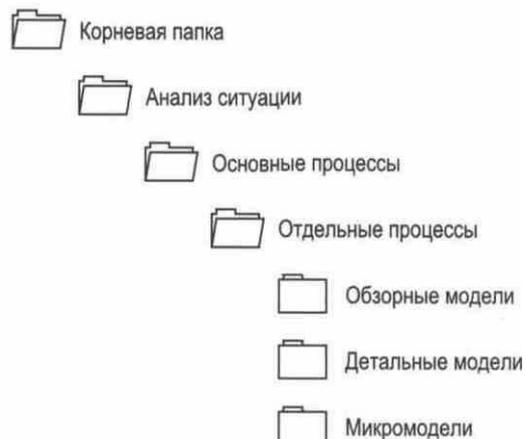
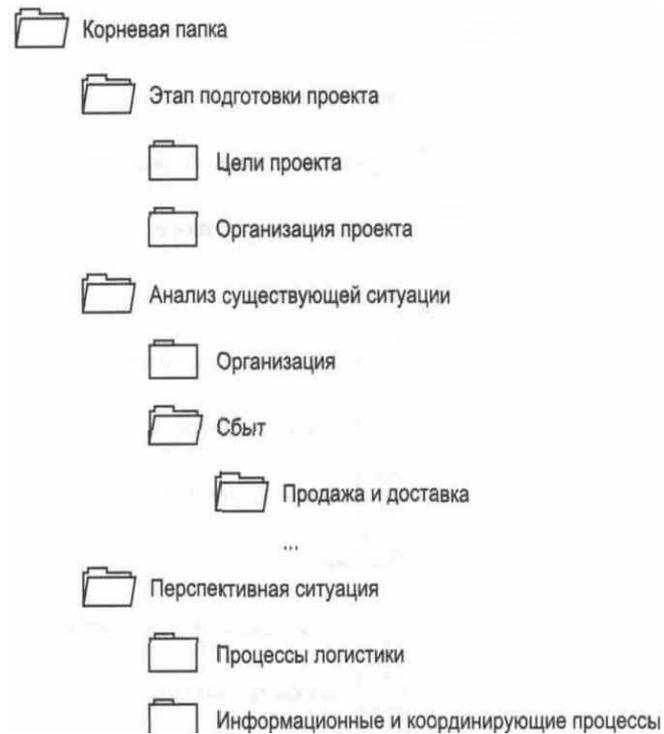


Рис. 183. Иерархия папок, построенная при комбинированном подходе «этапы моделирования процессный подход»



**Рис. 184.** Иерархия папок, построенная при комбинированном подходе «этапы моделирования - типы моделей, функциональная структура, процессный подход» (различные критерии на разных этапах моделирования)

## Общие принципы моделирования в ARIS

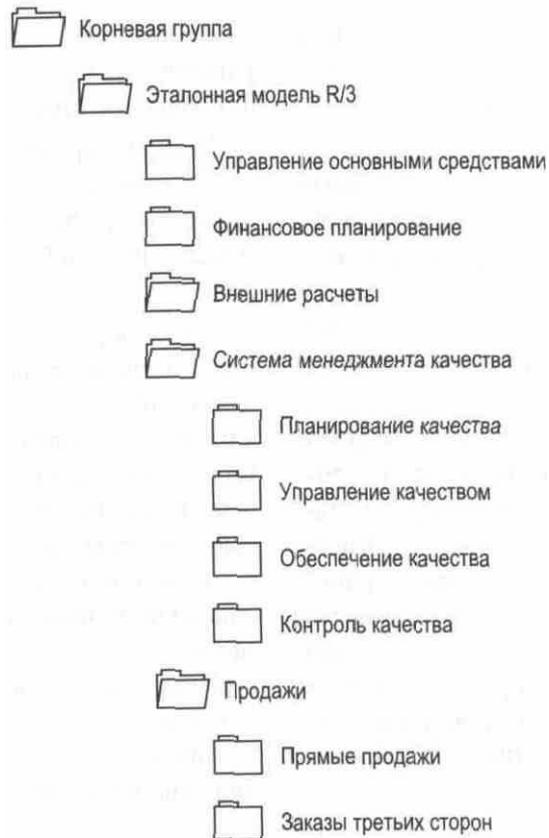


Рис. 185. Иерархия папок, построенная при комбинированном подходе «расширение структуры эталонной модели информационной системы при добавлении новых задач»

Иерархия папок позволяет навести порядок в создаваемых моделях и объектах и управлять доступом участников моделирования к отдельным его частям.

Однако часто становится недостаточным использование только одного иерархического дерева систематизации. Как правило, возникает необходимость представлять все создаваемые модели одновременно в нескольких разрезах — функциональном, процессном, в аспекте менеджмента качества и т. д.

Получить подобную картину можно с помощью специализированного типа модели — модели структуры (см. раздел 8.9), предназначенной для структуризации моделей организации. Эта модель строится из объектов одного типа — «структурный элемент». Каждый структурный элемент представляет собой критерий или значение критерия классификации моделей. Каждая модель может быть поставлена в соответствие одному или нескольким критериям. Тем самым

возможно формирование нескольких подходов к классификации моделей организации.

Между структурными элементами могут быть отношения, которые отражают как иерархию критериев, так и взаимосвязи между моделями, отнесенными к заданному критерию.

Например, в качестве критерия можно использовать функциональный подход, одним из элементов которого является область материального обеспечения компании (рис. 186). Материально-техническое обеспечение как макрофункция предприятия распадается на основные процессы, продукты и услуги, а также сферы ответственности. Далее следует иерархия описания — взаимосвязи между моделями: от сфер ответственности переходим к моделям структур процессов, которые, с свою очередь, приводят нас к описанию процессов компании. К этим же процессам можно прийти от описания основных процессов организации.

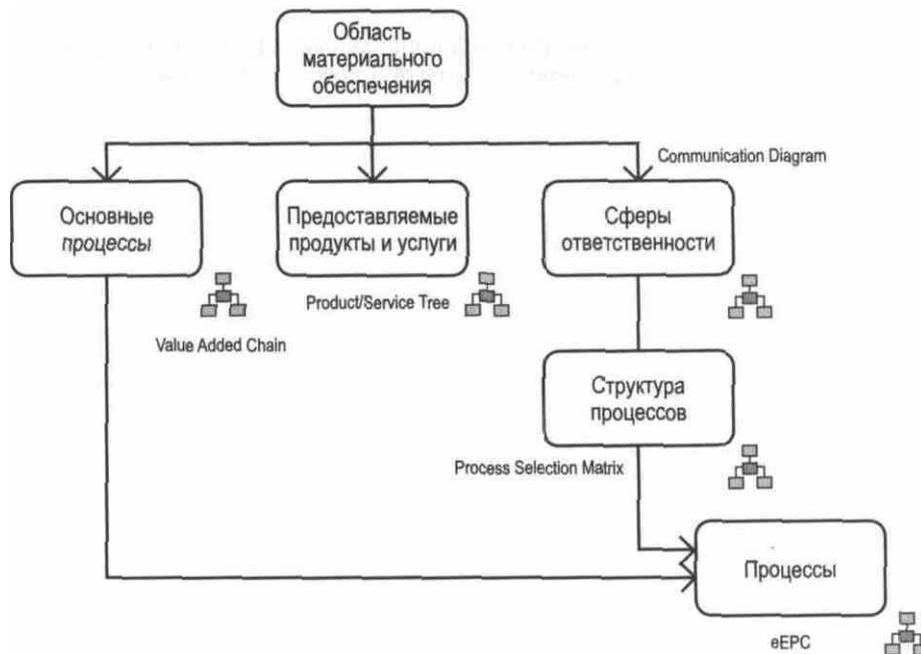


Рис. 186. Структурирование информации при помощи модели структуры

### Общие принципы моделирования в ARIS

## 9.3. Варианты моделей

Для целей создания и анализа альтернативных моделей используется концепция вариантов, подразумевающая создание вариантов моделей и объектов. С этой концепцией связаны такие понятия, как «мастер» (исходная модель) и «вариант» (получаемая на ее основе альтернатива). Каждая модель (объект) может иметь несколько вариантов, но только одного мастера. Концепция реализуется при помощи модуля *ARIS Variants* (см. раздел 5.17).

При создании варианта модели делается ее точная копия, состоящая из объектов, которые являются вариантами объектов мастер-модели. Впоследствии модель и каждый объект могут быть изменены независимо от мастер-модели и ее объектов, в том числе путем удаления объектов или создания новых. После внесения всех необходимых изменений может быть проведена процедура сравнения модели варианта с ее мастером. При этом сравнение может проходить по нескольким сценариям: проверка наличия или отсутствия объектов с различными сравниваемыми параметрами или сравнение характеристик объектов.

Таким образом, сравнение моделей может производиться на экспертном уровне, когда специалист в предметной области проводит визуальный анализ альтернативных моделей и исходя из своего опыта выбирает наиболее предпочтительный вариант.

Можно провести и более точное сравнение, если, например, задаться целью сократить времени продолжительности процесса. Для этого создается несколько вариантов исходной процедуры. Все эти варианты могут отличаться друг от друга:

- количеством исполнителей;
- наличием или отсутствием дублирования исполнителей;
- параллельными работами, выполняемыми различными исполнителями;
- графиками работ исполнителей и их дублеров и т.д.

Все разработанные варианты могут быть проведены через механизм динамического моделирования, который для каждой модели выдаст набор статистики как по процессу в целом, так и по отдельным его элементам. Статистика включает такие параметры, как среднее время выполнения процесса, общее время ожидания, среднее время выполнения отдельных функций, коэффициент использования исполнителей и других ресурсов и т. д. Полученная статистика может служить основой для сравнения альтернативных вариантов и выбора наиболее оптимального из них.

## 9.4. Отношения между моделями

Связи, или отношения, между моделями ARIS реализуется непосредственно через общие для них объекты.

Выделяют два вида отношений между моделями:

- использование одного и того же объекта в нескольких моделях;
- детализация объекта с помощью модели.

Для иллюстрации первого вида отношений можно привести следующий пример. Тип объекта *функция* существует как в модели типа eEPC, так и в модели типа *дерево функций*. То есть модель процесса связана с моделью иерархии функций через общие для них функции. При этом одни и те же типы объектов в различных моделях могут изображаться различными символами. Это необходимо для конкретизации предназначения типа объекта. Например, *функция* (в модели eEPC, функциональное дерево и т.д.) и *операция* (в UML-ди-аграмме класса) имеют различные обозначения, но оба понятия отображают один и тот же тип объекта. Таким образом, осуществляется интеграция подходов к моделированию деятельности.

При использовании одного и того же объекта в нескольких моделях каждая из них содержит его различные экземпляры. Атрибуты этих экземпляров идентичны во всех моделях. Создание нового экземпляра осуществляется посредством последовательности операций *Copy* (*Копировать*) и *Paste* (*Вставить*).

Детализация производится путем привязки модели экземпляру. Для этого следует воспользоваться командой *Assignment/Create* (*Детализация/Создать*) контекстного меню объекта или командой *New* (*Новая*) во вкладке *Assignments* (*Детализации*) диалогового окна свойств объекта.

#### Моделирование бизнеса. Методология ARIS

При этом детализация будет создана не для каждого экземпляра объекта, а для его описания. Иными словами, создается детализация одного из объектов модели, но она же будет проведена для всех экземпляров, в какой бы модели объект ни присутствовал.

Указанные два механизма формируют сложную систему взаимосвязей моделей, обеспечивающую комплексный взгляд на моделируемый объект как с точки зрения различных аспектов его функционирования, так и с точки зрения различных форм описания.

Примеры использования одного типа объекта в нескольких типах моделей представлены в табл. 45. Различные виды детализации объектов приведены в табл. 46.

**Таблица 45. Использование объектов в различных типах моделей**

Типы объектов в UML-диаграммах	Типы объектов в других моделях ARIS
Действие (UML-диаграмма действию)	Функция (еЕРС, дерево функций и т.д.)
Операция (UML-диаграмма класса, UML-диаграмма описания класса)	Функция (еЕРС, дерево функций и т.д.)
Исполнитель (UML-диаграмма использования приложений)	Тип сотрудника (еЕРС, организационная схема)
Состояние (UML-диаграмма состояний)	Состояние объекта, результат (еЕРС и т.д.)
Состояние объекта (UML-диаграмма действий)	Состояние объекта, результат (еЕРС и т.д.)

## 9.5. Определение глубины моделирования

Комплексное описание организации состоит не из одной модели, а из множества взаимосвязанных моделей различных типов. Сложные процессы не следует представлять одной детальной диаграммой, поскольку она крайне сложна для восприятия и анализа. Необходимо начинать с общего обзора процесса, каждый этап которого затем итеративно уточняется дополнительными моделями.

Удобство восприятия результатов обеспечивает модели верхнего уровня обзором моделируемой предметной области (например, диаграмма добавленного качества, коммуникационная диаграмма и другие).

Дальнейшая детализация объектов на модели более низкого уровня должна быть тщательно продумана и принята участниками моделирования.

Определение глубины моделирования связано с выбором:

Таблица 46. Детализация объектов

Типы объектов	Привязываемые модели
Класс	eEPC, eERM, модель технических терминов, модель данных IEF, OMT-модель описания класса, динамическая модель OMT, модель SAP-SERM, SeDaM, UML-диаграмма действий, UML-диаграмма описания класса, UML-диаграмма класса, UML-диаграмма состояний
Операция Деятельность Функция	eEPC, eEPC (с потоком материалов), дерево функций, диаграмма окружения функции, производственный процесс, диаграмма информационных потоков, диаграмма потоков материалов, офисный процесс, матрица выбора процесса, SAP ALE-модель фильтров, SAP ALE-функциональная модель, SAP ALE-модель потоков сообщений, SAP-диаграмма приложений, модель "системные атрибуты", UML-диаграмма действий, UML-диаграмма использования приложений, VAD-диаграмма цепочки добавленного качества
Состояние Состояние объекта Результат	eEPC, eEPC (с потоком материалов), дерево функций, производственный процесс, диаграмма движения продуктов/услуг, дерево продуктов/услуг, офисный процесс, матрица выбора продукции, дерево продуктов, диаграмма окружения продукта, матрица выбора процесса, UML-диаграмма состояний
Пакет	UML-диаграмма действий, UML-диаграмма описания класса, UML-диаграмма класса, UML-диаграмма взаимодействия, UML-диаграмма компонент, UML-диаграмма состояний, UML-диаграмма использования приложений
Исполнитель Тип сотрудника	Календарь смен
Решение Разделить Синхронизировать Правило	Диаграмма правил
Пример использования	eEPC, eEPC (с потоком материалов), UML-диаграмма действий, UML-диаграмма взаимодействий, UML-диаграмма использования приложений, PCD, PCD (с потоком материалов)

### Моделирование бизнеса. Методология ARIS

- степени обобщения и детализации разрабатываемых моделей;
- количества уровней моделей, представляющих переход от общих моделей к детальным.

Первый пункт связан с целями и задачами моделирования. Например, для внедрения информационной системы необходимо описание до уровня рабочих мест или даже до уровня операций на рабочих местах с построением моделей структур данных и т. п.

Второй пункт связан с необходимостью получения наглядных моделей, не перегруженных информацией. Для определения глубины моделирования вводится понятие иерархических и неиерархических моделей.

*Иерархические модели* — это модели, которые отражают подчинение объектов одного типа другому. К ним относятся диаграмма целей, дерево функций, организационная схема и другие. Для этих типов моделей необходимо определить количество отражаемых на одной диаграмме уровней иерархии. Как правило, диаграммы более чем с четырьмя или пятью уровнями иерархии становятся плохо читаемыми и тяжелыми для восприятия, а аккуратное размещение объектов в поле модели — проблематичным.

Для уменьшения количества уровней на одной диаграмме используется механизм детализации: объекты четвертого или пятого уровней детализируются моделью того же типа, где и отражаются последующие 4-5 уровней иерархии (рис. 187).

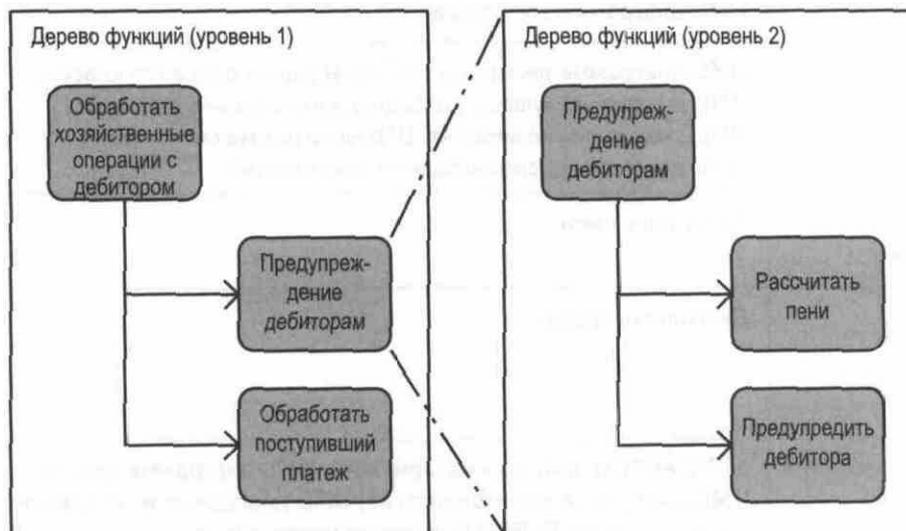


Рис. 187. Иерархические модели

*Неиерархические модели* — это модели, которые отражают взаимосвязь объектов одного уровня. К ним можно отнести модели процессов VAD, eEPC, PCD и другие. В этом случае необходимо управлять степенью загруженности модели — чем меньше количество уровней используется, тем более

детальными, а значит, и загруженными получаются модели.

Таким образом, формируется карта иерархии моделей — переход от моделей верхнего уровня обобщения к детальным моделям. Пример такой карты представлен на рис. 188.

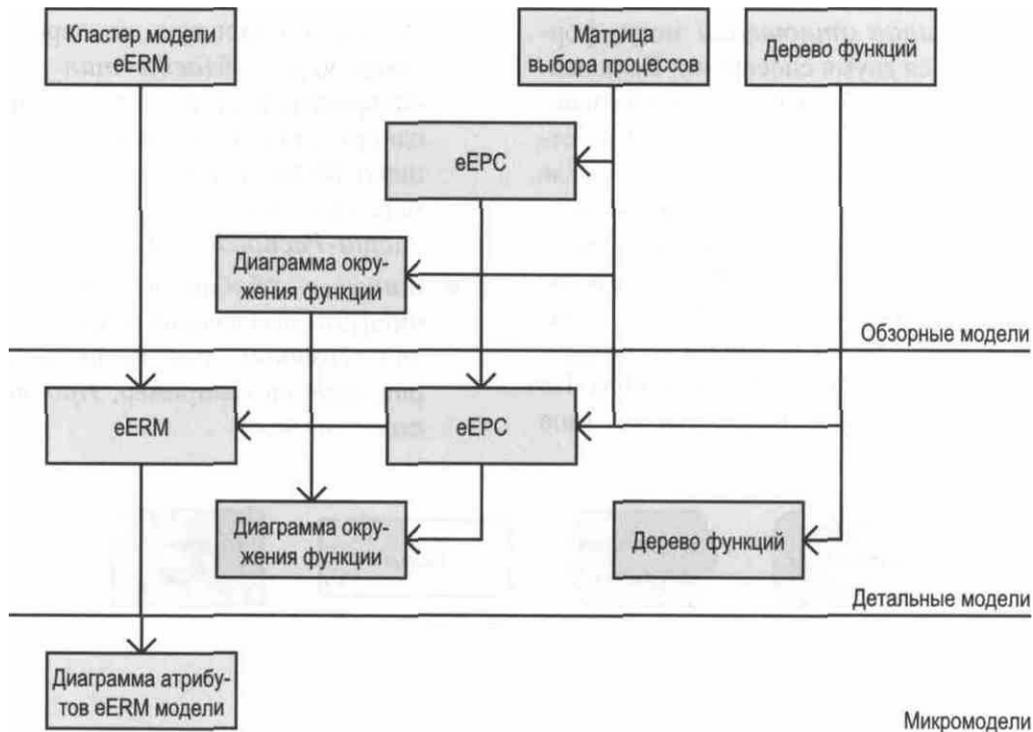


Рис. 188. Пример карты иерархии моделей

## 9.6. Правила именования и оформления моделей

### 9.6.1. Задание правил именования

В рамках модели организации объекты часто идентифицируются при помощи их имен. Строгие правила присвоения имен объектам обеспечивают целостность модели, удобство ее чтения, а также облегчают поиск объектов в базе. Правила именования должны быть определены для каждого типа объекта (рис. 189).

Существует два аспекта, которые следует учитывать в процессе задания правил именования объектов, — *синтаксический* и *семантический*.

Первый определяет формы используемых слов и словосочетаний, например:

- имя *функции* состоит из глагола, отражающего операцию, и существительного, указывающего на объект, над которым эта операция производится;

- имя *события* состоит из существительного, указывающего на объект, и глагола, отражающего операцию, которая была произведена над этим объектом;

- имя *технического термина* соответствует принятому в данной предметной области объекту (обычно используется существительное);

- имя *сущности* отражает описываемый ею объект предметной области, задается в виде единственного существительного, (например, *продукт* или *человек*). Составные термины формируются в соответствии с грамматическими правилами. Чтобы точнее определить существительное, можно дополнить его прилагательным или другим существительным через тире (например, *клиент-корпоративный* или *запас-продукт*);

- имя *кластера* состоит из существительного, характеризующего ограниченную область изучения сложного объекта. Это имя должно быть идентичным имени центрального типа сущности данного кластера;

имена *типов отношений* могут формироваться двумя способами. Если возможно, то желательно использовать независимый термин, производный из отношения. Например, *емкость* — для типа отношения между типами сущностей, *устройство размещения емкости* и *период времени*. Во всех других случаях имена формируются связыванием имен типов сущностей, находящихся в отношении. В отношениях 1:п имя иерархически подчиненного типа

сущности помещают на первое место (например, *область-тип\_области-распределение*). Если типы отношений одного ранга, то на первое место помещают технически более важное (например, *Тип\_области-Группа\_типа\_области-Распределение*),

*типы обобщения/специализации* именуют объединением имени супертипа с термином, ссылающимся на критерий деления (например, *Продукт-Использование*).

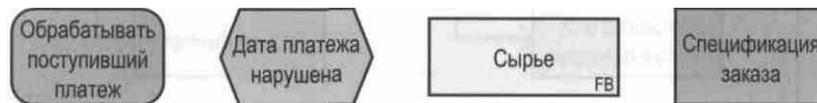


Рис.189. Имена объектов

Если выбранное имя объекта длиннее 40 символов, рекомендуется использовать логичное однозначное сокращение. В этом случае полное имя должно быть введено в атрибут объекта *полное имя {Full name}*.

Сам по себе синтаксический аспект именования не может гарантировать единства названий в проекте. Одно и то же действие может быть названо несколькими способами,

т.е. с помощью слов, близких по смыслу. Например, *создать заказ* и *составить заказ*.

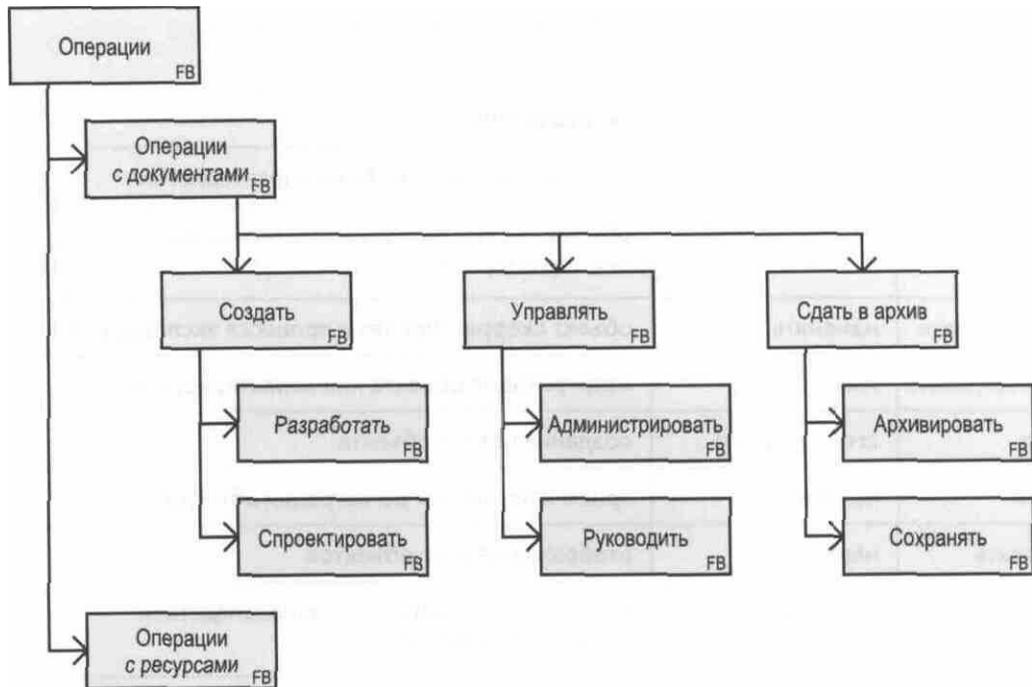
При решении семантической проблемы целесообразно составить глоссарий терминов для именования объектов. Пример такого глоссария приведен в табл. 47.

Для анализа и описания терминологии предметной области могут быть использованы модели технических терминов (рис. 190.)

## Общие принципы моделирования в ARIS

Таблица 47. Глоссарий терминов для именования объектов типа «функция»

Операция	Синонимы	Объяснение
регулировать	настраивать	настройка объектов или их характеристик
обновлять	модернизировать	обусловленная временем модификация объекта или характеристики
корректировать	изменять	объект скорректирован в процессе эксплуатации
модифицировать	изменять	модификация объекта или характеристики
создать	сгенерировать	создание нового объекта
принять	получить	прием и первичная регистрация объекта
отобразить	нарисовать	отображение всех объектов
архивировать	сохранять	сохранение объекта так, что впоследствии он мог быть извлечен
выбрать	указать	выбор объекта из некоторого списка
вычислить	рассчитать	рассчитать числовое значение
выполнить	исполнить	выполнение предварительно определенной деятельности
удалить	стереть	удаление объекта
найти	искать	обнаружение объектов/характеристик, удовлетворяющих определенным критериям
копировать	клонировать	повторное воспроизведение существующих значений/объектов без модификаций
переслать	передать	пересылка объектов для дальнейшей обработки
сгруппировать	объединить	связывание объектов согласно определенным критериям



**Рис. 190. Модель технических терминов**

Информация о «явных» объектах, таких как устоявшиеся формы рабочих документов (рис. 191), типовые отчеты, стандартные бланки и т.п., может быть занесена в базу данных еще до начала моделирования.

При этом все участники моделирования будут использовать уже занесенные объекты, а не создавать новые. Таким образом, снижаются трудозатраты на поддержание моделей в целостном виде.

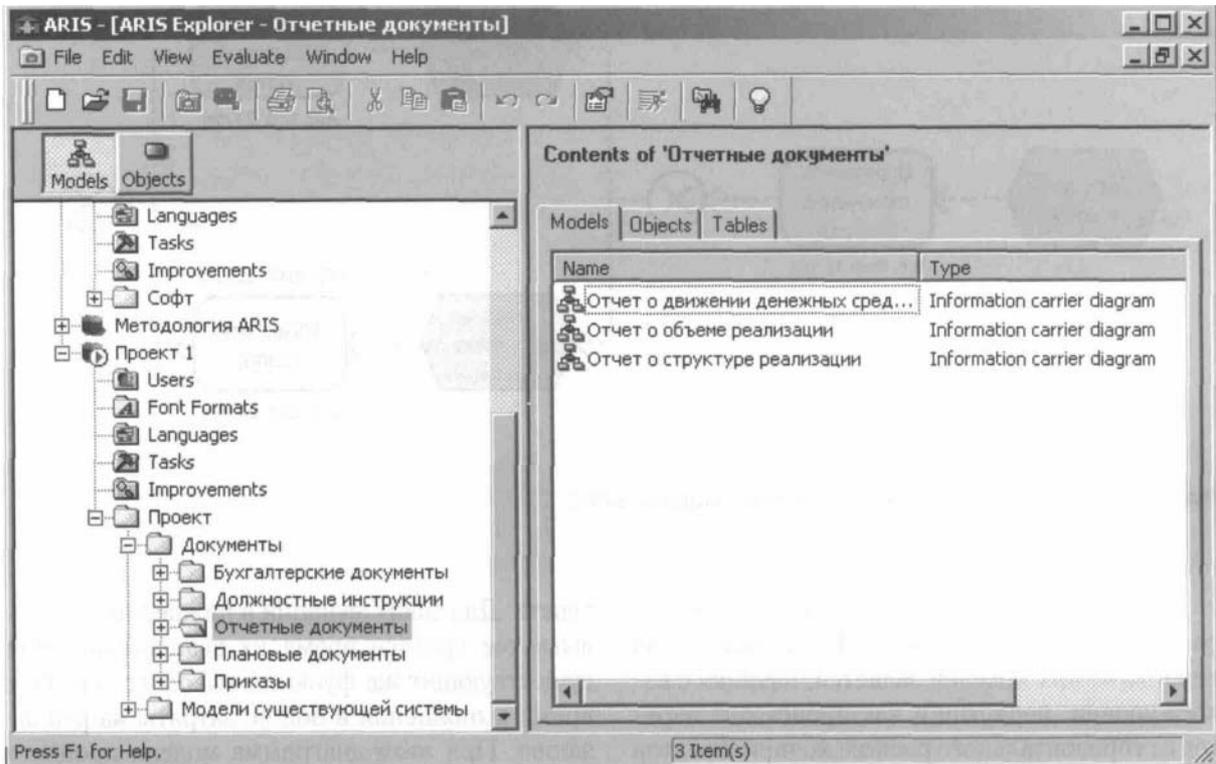


Рис. 191. Сохранение информации об устоявшихся формах рабочих документов

### 9.6.2. Задание правил графического оформления моделей

Одним из важнейших факторов успеха моделирования является получение наглядных моделей, которые с легкостью воспринимаются всеми заинтересованными лицами — технологами, специалистами из бизнес-подразделений, руководителями. Такое качество может быть достигнуто при выполнении правил оформления моделей.

К правилам графического оформления объектов и моделей относят:

- внешний вид объектов;
- соответствующее (взаимное) расположение объектов в модели;
- отображение важных характеристик объектов в поле моделирования.

Определение внешнего вида объектов (цвет, форма, элементы оформления) позволяет отобразить некоторые особенности каждого объекта. При этом визуальная фильтрация объектов по их предназначению является крайне важной для быстрого чтения и анализа модели. Дополнительная информация по каждому объекту (атрибуты объекта) также повышает наглядность.

Можно ввести тень объекта, изменить его размер, цвет заливки, тип, толщину и цвет ограничивающей линии (рис. 192). На модели рядом с каждым объектом можно поместить значения его произвольно выбранных атрибутов, управляя при этом шрифтом надписей и их центровкой. В то же время всегда можно вернуться к стандартным параметрам оформления (рис. 193), пометив поле *Default Values* (Параметры по умолчанию).



Рис. 192. Графическое представление модели eEPC

Также важно задавать правила взаимного расположения объектов. Например, для иерархических моделей задается, начиная с какого уровня иерархии и как происходит переход от горизонтального расположения объектов к вертикальному. Для моделей процессов определяется расположение диаграммы — горизонтальное или вертикальное.

Функции, внесенные в рамках предложений по реорганизации, можно отображать с использованием утолщенных линий и наклонного текста. Для таких функций в нижней части указывается среднее время их выполнения. Для

существующих же функций выводится среднее время выполнения и общие затраты на реализацию. При этом диаграмма модели процесса располагается слева направо, как показано на рис. 192.

Изменение внешнего вида объекта и вынесение атрибутов осуществляется в диалоговом окне свойств объекта — закладки *Внешний вид объекта* {*Object appearance*} (рис. 193) и *Расположение атрибутов* (*Attribute placement*).

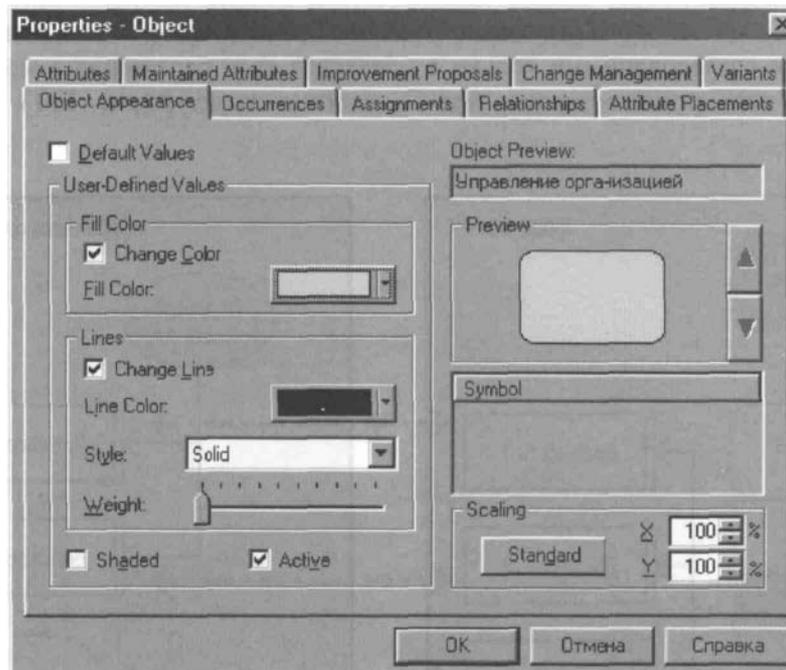


Рис. 193. Внешний вид объекта

ARIS имеет встроенный механизм шаблонов (*Templates*), позволяющий задавать внешний вид объектов и использовать его всеми участниками моделирования.

## 9.7. Проверка моделей

При моделировании необходимо соблюдать два вида правил — *синтаксические* и *семантические*.

К синтаксическим относятся общие правила использования и соединения объектов в моделях, а также правила детализации объектов в модели определенного типа. Например, возможно использование только одного типа связи между функцией и организационной единицей — «исполняет». Или модели процессов не должны включать составные логические операторы.

Синтаксические правила определяются методологией ARIS (метамодель ARIS), а также дополнительными настройками, ограничивающими применение этой методологии (например, настройка методологических фильтров). Проверка следования этим правилам проводится в процессе моделирования, т.е. система не позволит выполнить действие, не предусмотренное методологией (использование запрещенных объектов, связей, видов детализации).

Семантические правила связаны с синтаксическими правилами заполнения моделей. По семантическим правилам определяется степень полноты проработки и согласованность моделей и объектов. Например, модель процесса является полноценной, если для каждой функции этого процесса определен как исполнитель (тип связи «выполняет»), так и ответственный (тип связи «принимает решение»).

Модели процессов и организационных структур являются согласованными только в том случае, если все организационные элементы, представленные в моделях процессов, находят свое отражение в моделях организационных структур (рис. 194).

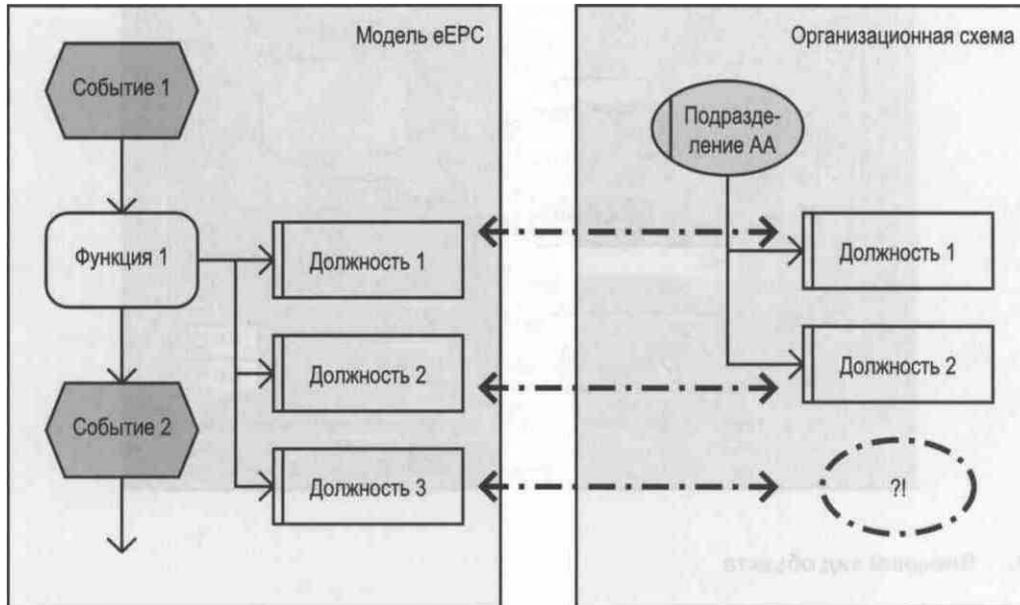


Рис. 194. Проверка полноты модели: организационная единица «Должность 3», присутствующая в модели eEPC, отсутствует в модели «Организационная схема»

Для проверки следования семантическим правилам используется механизм семантических проверок, реализованный в модуле *ARIS Semantic Check* (см. раздел 5.5). Проверки проводятся после завершения процесса моделирования.

В системе ARIS выделено семь основных типов правил семантических проверок:

- правила существования объектов (*Existence Rules*);
- правила взаимосвязи объектов (*Allocation Rules*);
- правила структуры моделей (*Structure Rules*);
- правила атрибутов объектов (*Object Attributes Rules*);
- правила атрибутов связей (*Relationships Attributes Rules*);

- правила детализации объектов (*Assignment Rules*);

- правила UML (*UML Rules*).

Все правила относятся к одному из указанных типов и для удобства группируются.

Система содержит набор правил, групп правил и типов правил. При этом для целей моделирования может быть создан специальный набор правил и их групп. Типы правил изменяться и создаваться не могут.

Ниже приведены некоторые правила, поставляемые с инструментальной системой ARIS.

Правила существования объектов

- Функция из модели процесса существует в дереве функций (целей).
- Функция из модели процесса существует в матрице выбора процесса.

- Функция из дерева функций не присутствует в другом дереве функций.

- Организационный элемент из модели процесса существует в организационной модели.

- Сущность из модели процесса существует в модели данных.

- Информационная функция из алгоритма модуля существует в структуре системы.

Правила взаимосвязи объектов

- Каждая функция должна иметь исполнителя.

- Каждое подразделение должно содержать в своем составе как минимум одну должность.

- Для каждой информационной системы должен быть определен ответственный за разработку.

Правила структуры моделей

С помощью этих правил проверяют корректность формирования структуры взаимосвязей объектов в модели. Правила объединены в несколько групп.

Общие структурные правила для всех типов моделей

- Каждый объект должен иметь одно или более соединений с другими объектами.

- Объект не может быть замкнут сам на себя.

Специальные структурные правила для всех моделей процессов

- Каждый путь должен начинаться и заканчиваться событием или интерфейсом в другой процесс.

- Проверка корректности количества входящих и исходящих соединений логических операторов.

- Все функции и события должны иметь только одно входящее и одно исходящее соединения.

- После события не должен следовать оператор «ИЛИ-НЕ» или «ИЛИ».

- В модели запрещены циклы.

- атрибутов для метода ABC.

Специальные структурные правила для моделей eERM

- Тип связи определяется по крайней мере для типов сущностей.

- Только один супертип представляет подтип.

- Группа атрибутов содержит не менее двух атрибутов.

*Специальные структурные правила для иерархических моделей* (относятся к иерархическим моделям с перекрестными связями и без них, например дерево функций, организационная схема и т.п.)

- Возможен только один корень.

- Каждый объект может иметь только одного родителя.

- Разрешено только одно соединение между двумя объектами.

- Все исходящие соединения объекта должны иметь один и тот же тип.

- Все соединения в модели должны иметь один и тот же тип.

Структурные правила для специфических иерархических моделей

- Только функции без исходящих соединений могут быть детализированы.

- Только организационные элементы без исходящих соединений могут быть детализированы.

Правила атрибутов объектов и связей

- Для всех функций заполнено поле описания.

- Для всех объектов заполнено поле имени.

- Для всех должностей заполнены возрастные требования.

- Для всех сотрудников заполнена дата рождения.

- Требования к заполнению атрибутов для динамического моделирования.

- Требования к заполнению

### Моделирование бизнеса. Методология ARIS

#### Правила детализации

Эти правила проверяют корректность детализации объектов с точки зрения используемых объектов и связей.

#### Детализация элементов данных

- Каждая сущность должна детализироваться в виде диаграммы окружения атрибутов.

- Каждый кластер должен детализироваться в виде модели данных.

#### Детализация функций

- Функция должна быть детализирована в виде дерева функции.

- Функция должна быть детализирована в виде диаграммы окружения функций.

Кроме того, существуют так называемые расширяемые правила проверки, которые легко изменять, используя закладку *Semantic Check* (Семантическая проверка), вызываемую через пункт меню *View/Option* (рис. 195). К ним относятся правила :

- существования объектов;
- взаимосвязей объектов;
- атрибутов объектов;
- атрибутов связей объектов;
- UML.

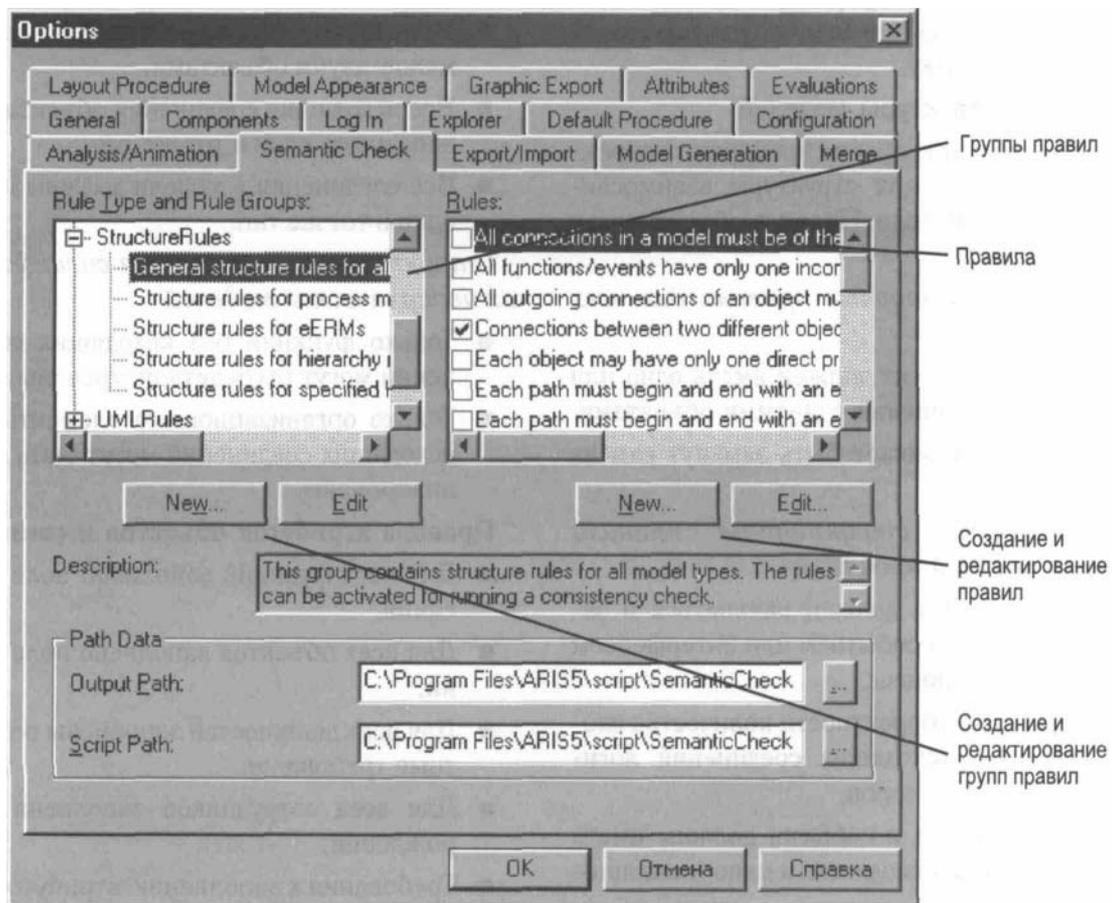


Рис. 195. Окно настройки правил семантической проверки (окно Options, закладка Semantic Checks)

Окно редактирования (создания) правил представлено на рис. 196. С помощью приведенных в нем инструкций можно осуществлять настройку правил всех типов, кроме *Assignment Rules* (Правила детализации) и *Structure Rules* (Правила

структуры). Эти типы правил не имеют графических мастеров, однако они могут быть скорректированы путем занесения записей в соответствующие файлы скриптов.

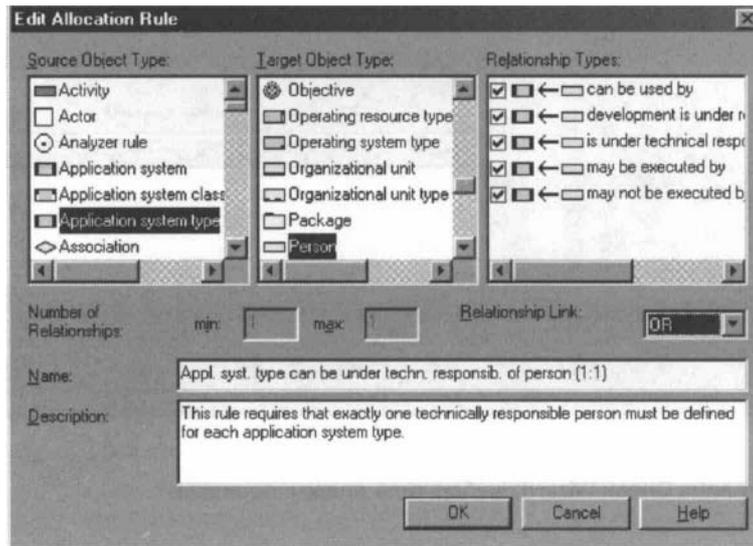


Рис. 196. Окно редактирования правил

Для проверки моделей и объектов механизм проверки семантики (*Semantic Check Wizard*) вызывается из пункта меню *Evaluate/Semantic Check*. Проверки проводятся

по набору правил, относящихся к заданной группе (рис. 197).

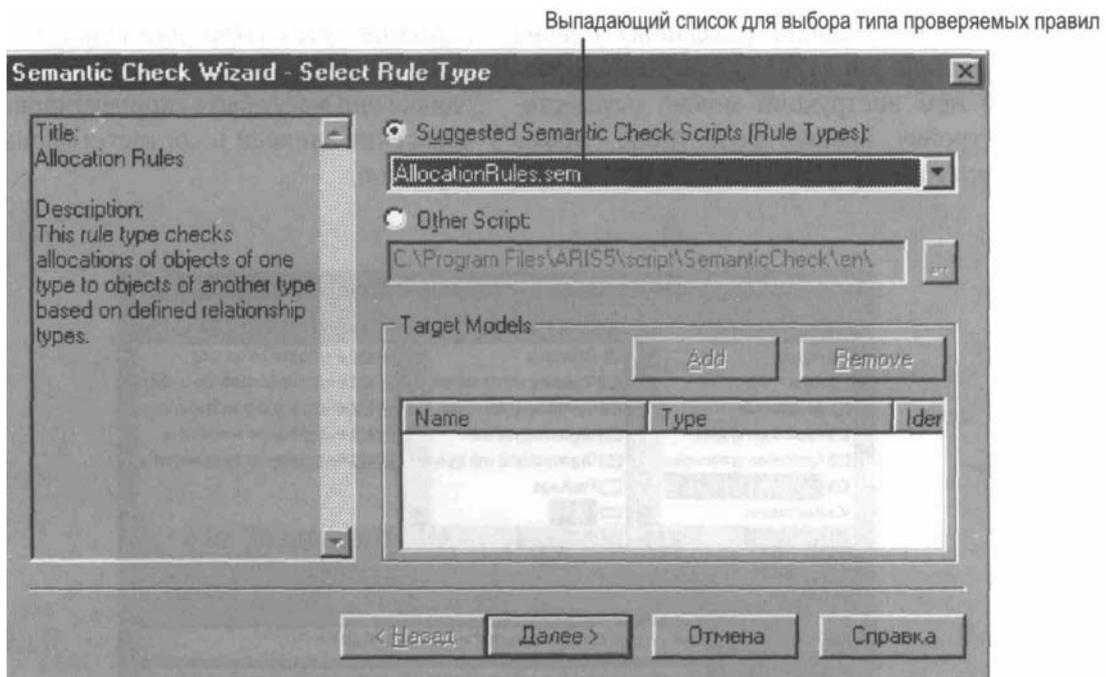


Рис. 197. Окно Semantic Check Wizard: выбор типа правил проверки

Отчет о результатах проверки семантики элементов ARIS может быть выведен в файл следующих форматов: *rtf*, *html*, *txt*, *doc*, *xls*. Файлы находятся в каталоге *C:\Program*

*Files\ARIS5\Script\Semantic Check\Out* и обычно называются *Semantic Check I.\**.

Страница типичного отчета о семантической проверке приведена на рис. 198.

	Allocation Rules	
<b>Rule Cluster is input AND output of a function</b>		
<b>Description:</b> The input and output cluster must be defined for every function.		
The following objects must be assigned to an object of type Cluster/Data model over all of the named relationships.		
Checked model	Object	Incorrect relationship
Обучение ARIS	Изучение ARIS	has output of
		is input for
	обучение	has output of
		is input for
	Сбор информации	has output of
		is input for
	Управление организацией	has output of
		is input for
Cluster is input AND output of a function		
25.12.00 15:58:08	C:\Program Files\ARIS5\script\SemanticCheck\out\Semantic Check1.rtf	Page 6 of 39

Рис. 198. Страница отчета о семантической проверке модели «Обучение ARIS»

## 9.8. Анализ моделей

Анализ является весьма важной частью методологии ARIS, позволяющей получить информацию об оптимальности моделей. Анализ реализуется при помощи модуля *ARIS Analysis* (см. раздел 5.7). Для проведения анализа могут применяться стандартные или разработанные пользователем сценарии (скрипты).

В *ARIS Analysis* возможны следующие стандартные скрипты для анализа моделей и объектов:

- организационные изменения в процессе (*Organizational Change in the Process*):

- информационные разрывы в процессе (*Media Breaks in the Process*),
- прикладные системы в процессе (*Application Systems in the Process*),
- классификация функций (*Function Classification*);
- сравнение моделей (*Model Comparison*),
- сравнение вариантов моделей (*Variant Comparison*).

Кроме этого, доступны еще два сценария для анализа моделей *BSC (Balanced Scorecard)*.

Правила проведения анализа моделей и объектов хранятся в каталоге *C:\Program Files\ARIS5\Script\Analyse\En* и имеют расширения *\*.asm* и *\*.aso*. Первые из них ориентированы на проверку моделей, вторые — на проверку объектов.

Скрипты анализа находятся в следующих файлах:

- *ApplicSysBreak.asm*. Используется для анализа следующих моделей: eEPC, eEPC (с потоком материалов), eEPC (в виде столбца и строки), производственный и офисный процессы, UML-диаграмма действий, PCD и PCD с потоком материалов. В результате анализа определяется коэффициент степени интеграции информационных систем внутри процесса. Он может принимать значения от 0 до 1. Чем ниже степень интеграции систем, тем выше индексный номер.

- *BSC\_GraphicExport.asin*. Используется для анализа BSC — причинно-следственной диаграммы, дерева функций, модели структуры, организационной схемы, VAD.

- *BSC\_PlanActualComparison.asm*. Применяется для анализа BSC — причинно-следственной диаграммы, дерева функций, модели структуры, организационной схемы, VAD. Выводит результаты сравнения фактических и планируемых показателей достижения целей компании.

- *FunctionClass.asm*. Применяется для анализа всех моделей, содержащих объекты типа *функция*. Проводится группировка всех функций выбранных моделей согласно значению атрибута определенного типа.

- *MediaBreak.asm*. Используется для разновидностей моделей eEPC, производственного и офисного процессов, UML-диаграммы действий, PCD и PCD с потоком материалов. В ходе анализа определяется коэффициент, отражающий степень информационных разрывов в процессах. Этот коэффициент может принимать значения от 0 до 1. Чем ниже степень информационной интеграции в процессе, тем выше значение коэффициента.

- *ModelComparison.asm*. Применяется для сравнения по крайней мере двух моделей. Анализ проводится двумя способами. Первый — сравнение существования, в ходе которого выявляется наличие или отсутствие одинаковых объектов в сравниваемых моделях. В качестве критериев идентичности могут выступать некоторые атрибуты, такие как имя, идентификатор, описание объекта, варианты объекта. Второй — сравнение атрибутов, при котором сравниваются атрибуты объектов с идентичными именами, находящимися в различных моделях.

- *OrgChange.asm*. Применяется для разновидностей модели eEPC, производственного и офисного процессов, UML-диаграммы действий, PCD и PCD с потоком материалов. Определяется степень ориентированности на процесс. Расчет основывается на количестве переходов ответственности от одного исполнителя к другому.

- *VariantComparison.asm*. Применяется для сравнения только двух моделей — мастер-модели и ее варианта.

При проведении анализа выполняются инструкции последовательно появляющихся окон *Analysis Wizard* (рис. 199 — 201). Первое окно запускается из пункта *Evaluate\Analysis* главного меню.

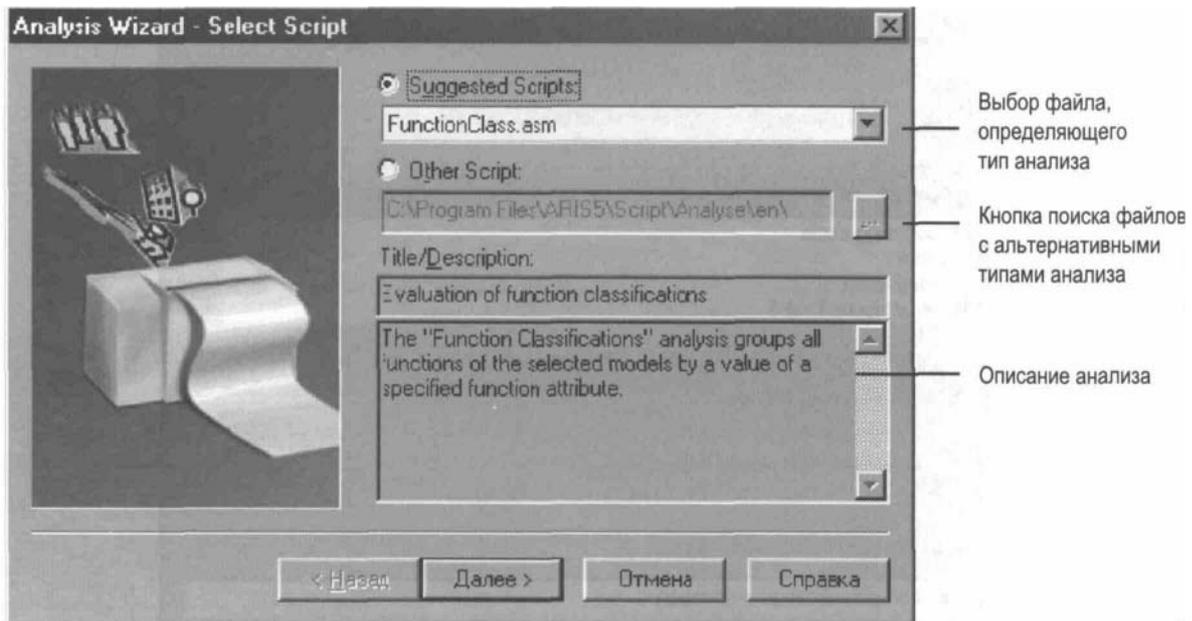


Рис.199. Первое окно Analysis Wizard: выбор скрипта

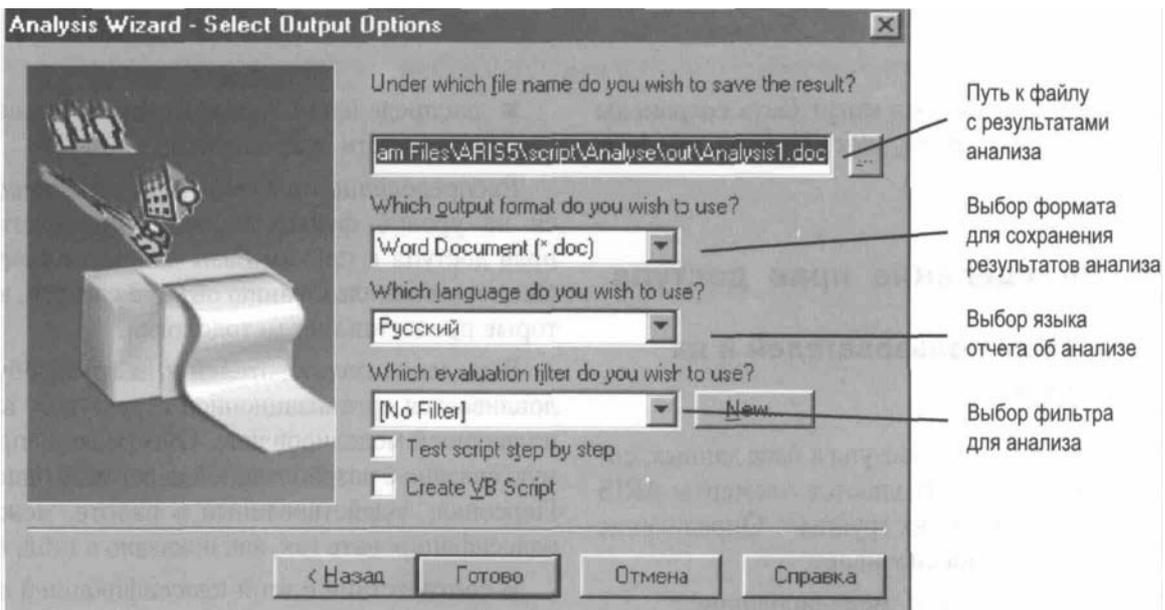


Рис. 200. Второе окно Analysis Wizard: выбор опции вывода информации

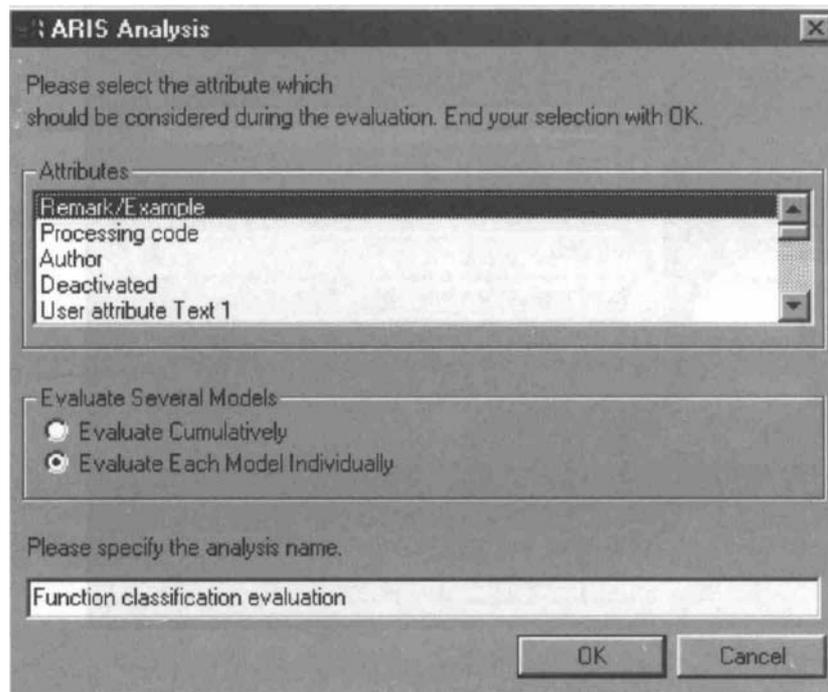


Рис. 201. Окно Analysis Wizard: выбор параметров анализа

Результаты анализа могут быть сохранены в форматах *.txt*, *.html*, *.doc*, *xls* или *.atf* (ARIS tableformat).

## 9.9. Определение прав доступа

### 9.9.1. Виды пользователей и их права

Для организации доступа к базе данных, содержащей модели, создаются элементы ARIS «пользователи» и «их группы». Определение этих элементов обеспечивает:

- распределенное моделирование;
- структурированный контролируемый доступ к моделям организации;

- распределение полномочий и ответственности участников.

Распределение полномочий осуществляется на уровне функциональных привилегий, прав доступа к папкам базы данных и разрешенного к использованию объема средств, которые предоставляет методология.

Выделение пользователей и их групп обуславливается организационной структурой, выполняющей моделирование. Оно также напрямую связано с разработанной иерархией папок. Персонал, задействованный в работе, можно классифицировать так, как показано в табл. 48.

В соответствии с этой классификацией отдельные сотрудники имеют различные задачи и права.

Таблица 48. Классификация пользователей ARIS

Должность	Служебное имя	Выполняемые функции
Администратор сервера	DBADMIN	Техническое администрирование
Администратор конфигурации	CFGADMIN	Конфигурационное администрирование
Администратор базы	System	Организационное администрирование
Руководитель моделирования	System Manager	Организационное администрирование, бизнес-администрирование
Пользователь	Конкретное имя -Victor, Alex	Выполнение отдельных работ с правами, настраиваемыми индивидуально
Группа пользователей	User Group или конкретное имя	Выполнение отдельных работ с правами, настраиваемыми индивидуально
"Гость"	Guest	Просмотр моделей без права внесения изменений

*Администратор сервера* — это пользователь с особыми привилегиями. Он является единственным пользователем, отвечающим за администрирование на сервере. В частности, он может создавать, реорганизовывать, удалять, сохранять и восстанавливать базы. Системное имя администратора сервера — DBADMIN, и оно не может быть изменено. По умолчанию администратор сервера получает пароль «DBADMIN». Этот пароль может быть изменен через контекстное меню сервера (диалоговое окно *Properties (Свойства)*, закладка *Administration (Администрирование)*).

В задачи администратора сервера входят:

- установка и инсталляция базы данных сервера;
- установка программного обеспечения сервера;
- создание новых баз данных;
- резервное копирование баз данных;
- восстановление после сбоев;

- конвертирование баз данных для обеспечения совместимости с новыми версиями инструментальной системы ARIS;
- создание баз данных пользователей.

*Администратор конфигурации* — это пользователь тоже с особыми привилегиями. Он является единственным пользователем, отвечающим за конфигурирование базы на сервере. Администратор может создавать, редактировать и удалять фильтры конфигураций, шаблоны и форматы шрифтов в базе конфигураций. Системное имя администратора конфигурации — CFGADMIN, и оно не может быть изменено. По умолчанию администратор конфигурации получает пароль «CFGADMIN», причем этот пароль может быть изменен через контекстное меню сервера (диалоговое окно *Properties*, закладка *Administration*).

*Администратор базы* — это особый системный пользователь. Он создается в каждой новой базе и не может быть удален. Системный пользователь — это тип пользователя в среде ARIS, который имеет все права на базу данных.

В задачи администратора базы входят:

- формирование иерархии папок;
- управление пользователями;
- управление правами доступа;
- управление языками, шрифтами.

Администратор базы является исполнителем решений руководителя проекта по моделированию, который в случае необходимости может самостоятельно осуществлять настройки.

В задачи руководителя проекта по моделированию включаются:

- администрирование процесса моделирования;
- определение структуры модели организации и прав доступа;
- выбор методов моделирования;
- определение правил автоматической идентификации объектов;
- принятие решений о форме отчетности;

- мониторинг консолидации объектов и моделей.

*Рядовые пользователи* осуществляют работы по моделированию в соответствии с их правами, выделенными с учетом их роли.

Для удобства администрирования пользователи объединяются в *группы пользователей*. Каждый пользователь имеет набор прав, образованный на основе его личных прав и прав, полученных посредством вхождения в одну или несколько групп пользователей.

Рекомендуется также выделить и настроить среди пользователей так называемого гостя, который имеет права навигации по моделям без возможности их изменения.

По умолчанию задается администратор сервера, администратор конфигурации и администратор базы. Остальные пользователи должны быть введены в базу данных посредством механизма управления пользователями (рис. 202.)

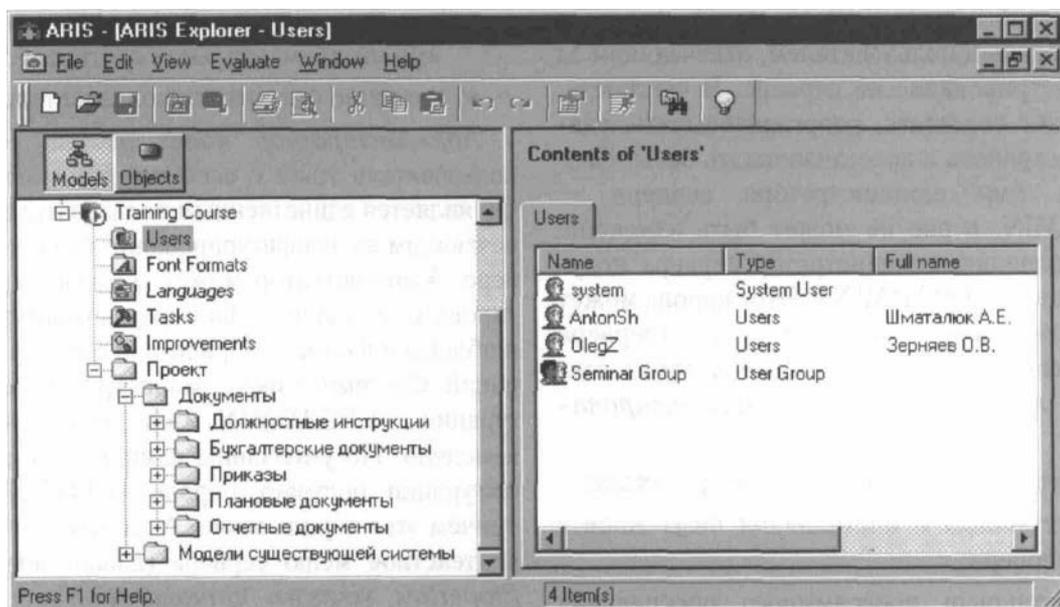


Рис. 202. Закладка Users

## Общие принципы моделирования в ARIS

Каждый пользователь имеет свои индивидуальные настраиваемые свойства:

- методологические фильтры (*Method Filter*),
- атрибуты (*Attributes*),
- префиксы идентификации (*Identifier*)
- пароль (*Password*);
- вхождение в группы пользователей (*User Group Association*);
- функциональные привилегии (*Function Privileges*).

Свойство «функциональные привилегии» позволяет разрешать и запрещать выполнение следующих действий в рамках базы данных (рис. 203):

- управлять изменениями (*Change management*);
- экспортировать данные из моделей (*Database export*);
- управлять базой данных (*Database management*);
- управлять шрифтами (*Font format management*),
- изменять нотации моделей (*Method changes*);
- управлять идентификацией (*Prefix management*),
- просматривать существующих пользователей (*Show user management*);
- управлять пользователями (*User management*).

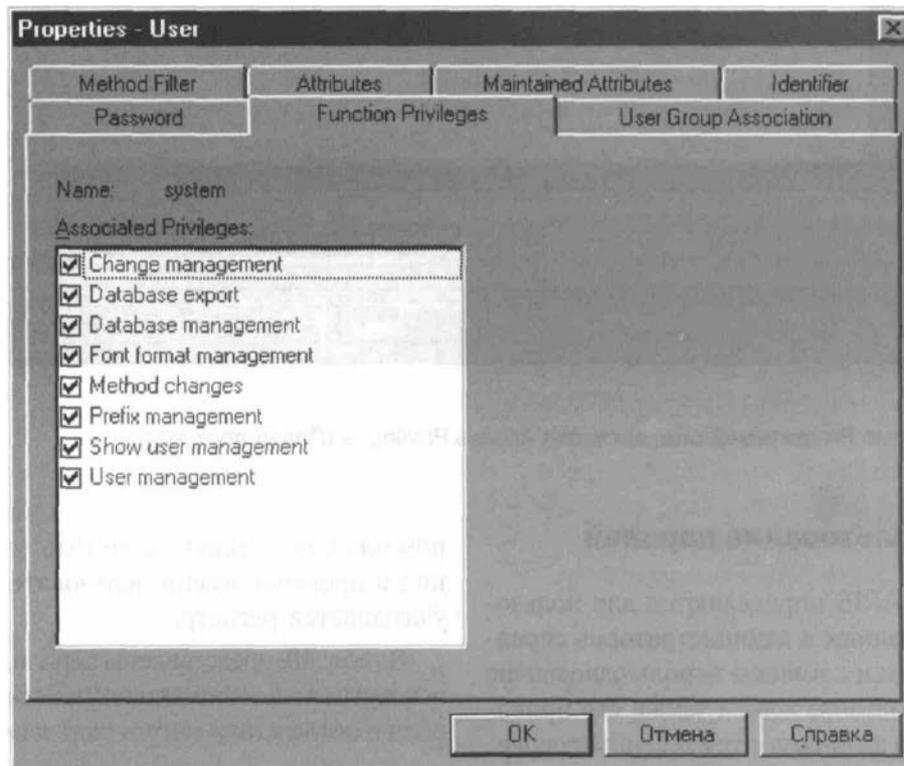


Рис. 203. Окно Properties-User, закладка Function Privileges (Функциональные привилегии)

### Моделирование бизнеса. Методология ARIS

Права доступа отдельных пользователей или их групп к моделям и объектам могут быть определены через свойства папок, содержащих эти модели и объекты (рис. 204). Закладка *Access Privileges (Права доступа)* позволяет установить следующие права:

- нет доступа — *No access* {—};
- чтение — *Read*(r—);
- чтение и запись — *Read+Write* (rw-);
- чтение, запись и удаление — *Read+Writ+Delete* (rwd).

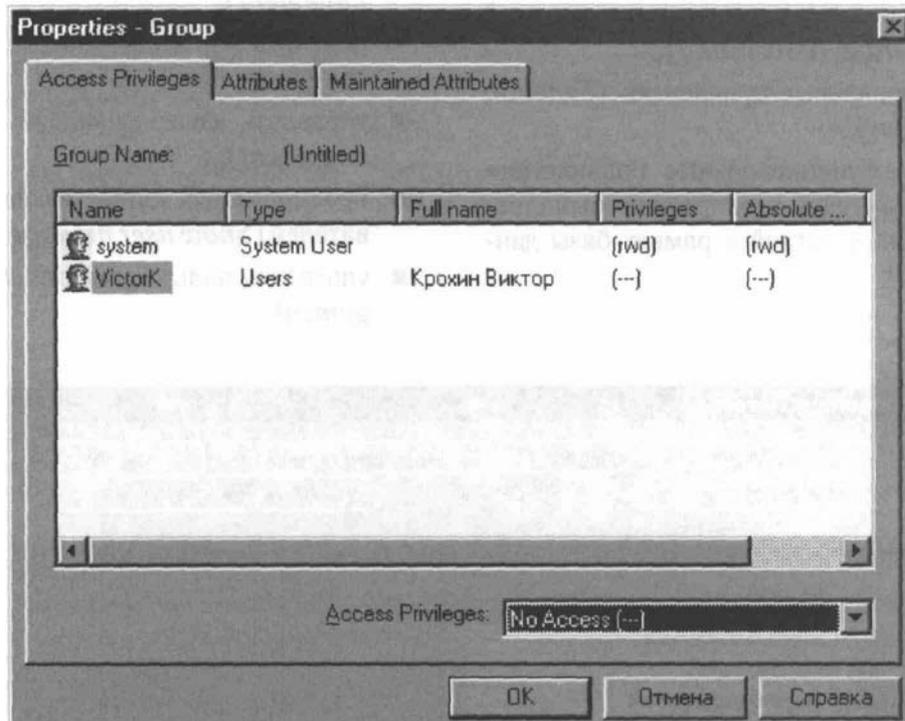


Рис. 204. Окно *Properties-Group*, закладка *Access Privileges (Права доступа)*

#### 9.9.2. Использование паролей

Пароли в ARIS определяются для пользователей баз данных и администраторов серверов. В сочетании с именем пароль однозначно определяет пользователя, а также его привилегии — права на работу с тем или

иным сервером или базой. Важно отметить, что при задании и проверке имени пользователя и пароля учитывается регистр.

В табл. 49. представлены пароли, задаваемые основным пользователям по умолчанию. Эти пароли в последствии могут быть изменены.

Таблица 49. Пароли, задаваемые пользователями по умолчанию

Имя	Пароль	Примечание
DBADMIN	DBADMIN	Администратор сервера. Задается в свойствах сервера.
CFGADMIN	CFGADMIN	Администратор «конфигурации». Задается в свойствах сервера.
system	Manager	Администратор базы. Задается в закладке управления пользователями базы данных.

ARIS позволяет настроить стандартные пароли, которые следует использовать при работе с системой. Настройка осуществляется посредством закладки *Log in (Соединение)* диалогового окна, вызываемого через меню *View/ Options*. Содержащиеся в этом окне значения будут автоматически подставляться системой при попытке доступа к ресурсам ARIS. Например, при открытии базы данных будут автоматически подставляться значения из раздела *User Defaults* (имя пользователя базы данных, его пароль и используемый методологический фильтр). В том случае, если подставляемые значения не соответствуют существующим (например, в базе нет такого пользователя или заданный пароль неверен), то система предложит вручную ввести эти значения.

### 9.9.3. Необходимость управления доступом

Среда ARIS имеет развитую систему доступа к базам данных моделей. Это означает, что возможна организация структурированного доступа участников к

информации по моделям, а также исключение «нежелательных» посетителей.

Более того, за счет контролируемого распределения такой функциональной привилегии, как экспорт данных, возможно исключение утечки информации при несанкционированном ее использовании.

Таким образом, система ARIS как инструментарий для комплексного моделирования и анализа деятельности организации заставляет всерьез задуматься о подготовке процесса моделирования с точки зрения методологии, правил проверки, распределения доступа.

Игнорирование данного этапа с большой вероятностью приведет к получению непригодного для анализа материала в виде неполноценных и несогласованных моделей, которые в этом случае не будут отличаться от других неформализованных способов отражения информации.

Только грамотная подготовка и контроль позволят получить результаты моделирования, которые можно будет эффективно использовать в дальнейшем.

# 10. Документирование моделей

## 10.1. Общие принципы документирования

Помимо графических моделей, конечным продуктом моделирования должен стать набор документации по проведенным работам. Комплект документов должен в удобной для ознакомления форме представлять всю важную для пользователя информацию. С другой стороны, документация должна служить исходным материалом для дальнейших работ — последующих этапов моделирования, тестирования и использования полученных решений.

С точки зрения проекта, связанного с моделированием, документирование — это вывод представленной в моделях информации в виде текстовых описаний, содержащихся в файлах заданного формата. Целями создания документов - отчетов являются:

- структурирование и обобщение информации о моделируемой деятельности;
- получение информации о проекте без привлечения специализированных графических средств;
- представление информации о проводимых работах в форме, удобной для различных групп пользователей;
- получение документов, связанных с закрытием этапов работ;
- формирование технологических и регламентных документов (положения о подразделениях, должностные инструкции, технологические карты и т.д.).

Структуры документов должны быть разработаны для всех групп заинтересованных лиц:

- руководителей организации;
- заказчиков проекта по моделированию бизнеса;

- рабочих групп моделирования по различным направлениям: информационное моделирование, описание структуры документов, описание организационной структуры, создание моделей верхнего уровня, описание процессов;

- руководителей проекта по созданию моделей деятельности;

- сотрудников подразделений, участвующих в проекте;

- клиентов организации и средств массовой информации.

Важно отметить, что отчеты не следует воспринимать исключительно как альтернативу графическим моделям. В большинстве случаев графическое представление информации является значительно более наглядным и удобным для использования материалом, чем его текстовая копия. Представленная текстовая информация должна дополнять свой графический источник.

Например, просматривая некоторую модель процесса, можно увидеть формирующиеся в нем рабочие документы. Однако из этой модели не видны все возможные пользователи сформированных документов, которые «разбросаны» по множеству моделей процессов. Специальным образом же созданный отчет позволит получить не только информацию по рассматриваемой модели процесса, но и дополнительные сведения в виде перечня лиц, использующих сформированные документы.

ARIS позволяет вывести в отчетный документ любую информацию, содержащуюся в базе данных проекта: описание моделей (включая графику), описание объектов с их характеристиками, взаимосвязи объектов не только в рамках выделенных для документирования моделей, но и по всей базе данных.

Процесс создания текстовых отчетов в ARIS полностью автоматизирован. Для этого используются так называемые скрипты — программы на языках *Visual Basic* и *ARIS Basic(G) Language Capability*, позволяющие перенести информацию из графических моделей в текстовые файлы в соответствии с определенными правилами. Под управлением скриптов проходит генерация отчета — создание текстового файла и запись его на магнитный диск.

Для предоставления пользователю базы данных доступной информации в форме таблицы или текста происходит обращение модулей ARIS к соответствующим скриптам. Скрипты создаются при помощи модулей *ARIS Script Editor* и *ARIS Basic(C) Language Capability* и могут использоваться модулями *ARIS Report*, *ARIS Analysis* или *ARIS Semantic Check* для анализа и оценки содержимого базы данных с точки зрения различных критериев. Существуют три основных типа скриптов, применяемых в различных модулях и хранящихся в файлах с разными расширениями: *ARIS Report* — .rs?, *ARIS Analysis* — .as?; *ARIS Semantic Check* — .se?

Дополнительным средством управления составом выводимой информации являются фильтры обработки — перечень типов моделей, объектов, связей, атрибутов, которые могут быть обработаны в процессе создания отчета. Все остальные элементы ARIS, не вошедшие в используемый фильтр обработки, игнорируются в процессе генерации отчета.

## 10.2. Генерация отчета по заданному скрипту

Для того, чтобы создать отчет по готовому скрипту, пользователь должен выполнить ряд действий при помощи модуля *ARIS Report*. Этот модуль представляет собой инструмент для анализа содержимого базы данных ARIS. Результаты анализа могут быть сохранены в

текстовом виде в наиболее распространенных форматах файлов HTML, DOC, RTF, XLS, TXT. Сохраненные результаты анализа называются *отчетами*. С помощью соответствующих приложений отчеты могут быть отредактированы и проанализированы.

Для генерации отчета прежде всего необходимо подключиться к базе данных, содержащей исходную информацию для будущего отчета. Далее следует выделить те элементы базы данных, информация о которых должна попасть в создаваемый отчет. Важно отметить, что все выделяемые элементы должны относиться к одному классу.

Отчет может генерироваться для следующих элементов ARIS:

- база данных;
- фильтр;
- папка;
- модель;
- объект (описание и экземпляры);
- таблица функционально-стоимостного анализа (модуль *ARISABC*).

Выбор элементов осуществляется в соответствии с правилами интерфейса операционной системы Windows. Особенностью системы ARIS является то, что отчет можно получить практически из любого модуля ARIS. При создании отчета система учитывает права доступа к содержащейся в базе данных информации. Пользователь должен иметь по крайней мере доступ к чтению выбранного содержимого базы данных.

После выделения интересующих элементов базы данных необходимо выбрать пункт меню *Evaluate/Report*, в результате чего будет запущен механизм создания отчетов, и на экране появится первое *окно ARIS Report Wizard (Создание отчетов ARIS)*. В нем нужно выбрать скрипт, который будет использован для создания отчета. При этом система автоматически предложит известные ей скрипты, подходящие для обработки выделенных элементов.

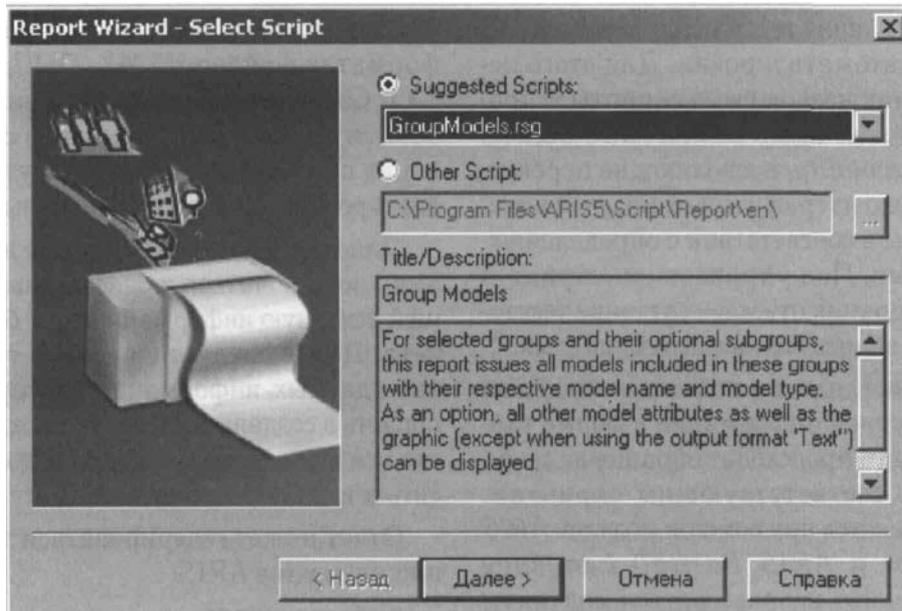


Рис. 205. Окно Report Wizard: выбор скрипта для генерации отчета

Предлагаемые системой скрипты будут отображены (рис. 205) в поле *Suggested Scripts* (*Предлагаемые скрипты*). Для поиска в файловой системе другого скрипта предназначено поле *Other Script* (*Другой скрипт*).

На следующем шаге (рис. 206) осуществляется стандартная настройка отчета, подразумевающая установку перечисленных ниже параметров:

- полное имя файла, в котором будет сохранен отчет (поле *Under which file name do you wish to save the result?*),

- формат, в котором будет сохранен отчет, - DOC, XLS, RTF, HTML или обычный (ASCII) текстовый файл (поле *Which output format do you wish to use?*);

- язык создаваемого отчета (поле *Which language do you wish to use?*). Система ARIS предоставляет возможность описывать модели одновременно на нескольких языках, и данная опция позволяет выбрать интересующий язык;

- в случае необходимости указывается фильтр обработки, сквозь который скрипт будет «видеть» содержимое базы данных ARIS (поле *Which evaluation filter do you wish to use?*).

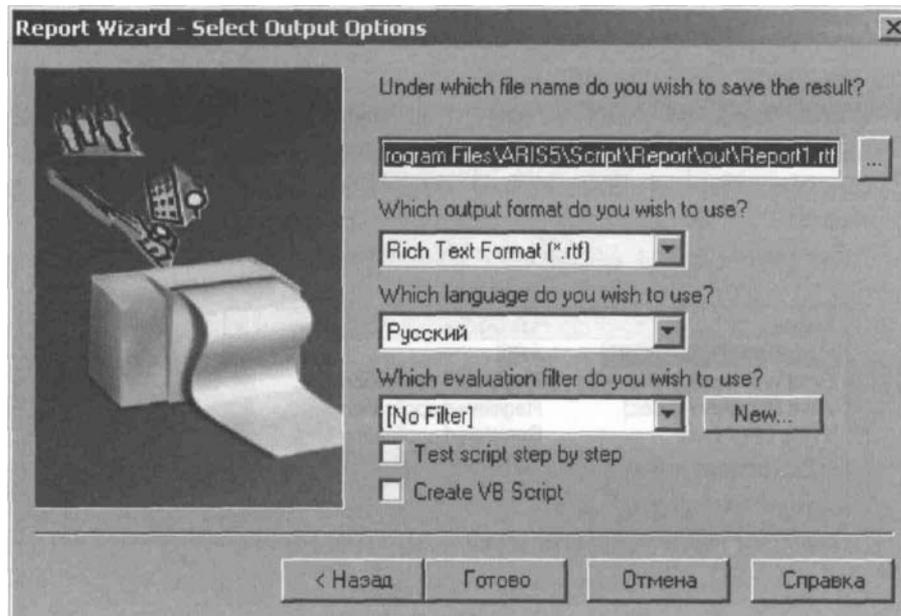


Рис. 206. Окно Report Wizard. Установка параметров отчета

Настройка — последний шаг в создании отчетов, после которого система запускает выбранный скрипт. Далее, в зависимости от используемого скрипта, могут быть выведены некоторые диалоговые окна для занесения дополнительных параметров формирования отчета.

Результатом работы скрипта становится документ в выбранном формате, содержимое которого система предлагает просмотреть после окончания процесса его формирования.

Каждый стандартный (поставляемый с системой) отчет состоит из общей части и раздела, содержащего специфичную для каждого конкретного скрипта информацию.

В общей части указываются:

- дата и время создания отчета;
- наименование сервера, содержащего базу данных, по которой был сгенерирован отчет;
- имя базы данных;
- имя пользователя, сгенерировавшего отчет.

Основные настройки модуля *ARIS Report*, используемые при генерации отчетов, производятся в меню *View/Options* с помощью закладки *Evaluations*. Вид этой закладки приведен на рис. 207.

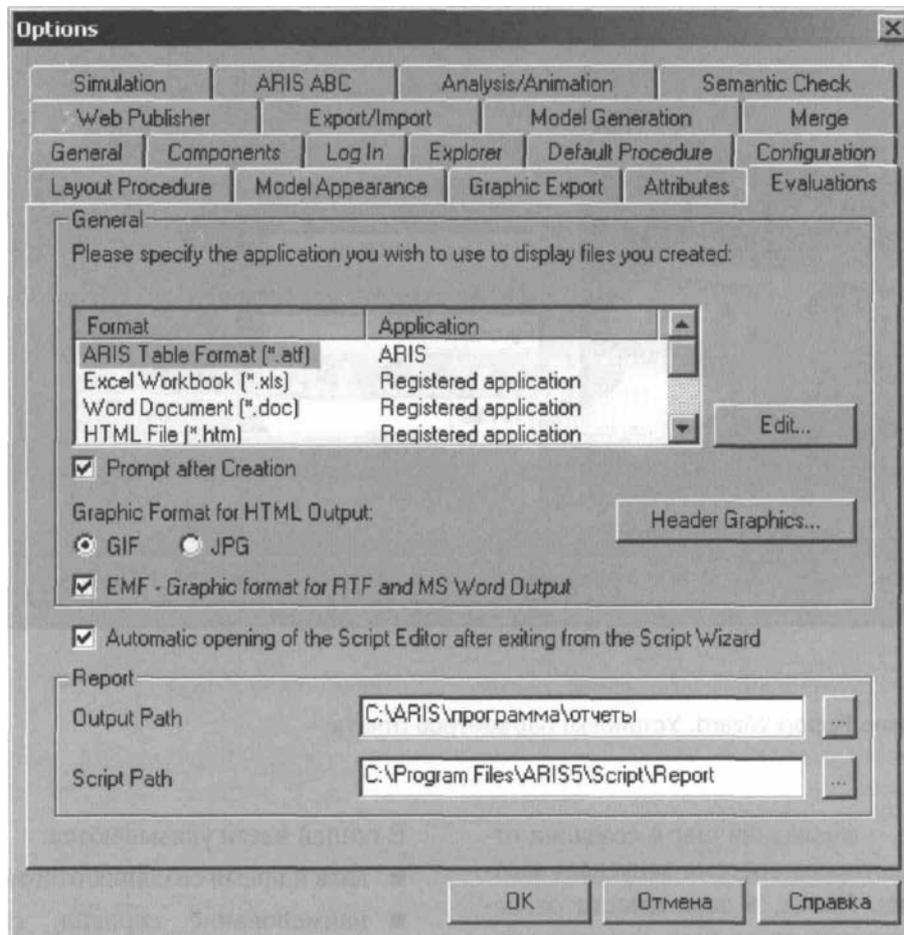


Рис. 207. Основные параметры для генерации отчетов

Область *General* (Общие установки) позволяет задать типы приложений, которые по умолчанию используются для просмотра и редактирования сгенерированных отчетов, а также определить формат представления графики при выводе отчета в файлы формата HTML.

В области *Report* (Отчет) задается путь, по которому система ищет скрипты для генерации отчета. Именно так выбираются скрипты в поле *Suggested Scripts* (Предлагаемые скрипты), показанном на рис. 205. Задается также путь к папке, в которую по умолчанию система записывает все генерируемые отчеты.

Кнопка *Header graphics* (Графика колонтитулов) открывает окно (рис. 58), в котором пользователю предоставляется возможность изменить логотипы, печатающиеся в верхнем колонтитуле отчетов справа и слева (рис. 216).

## 10.3. Стандартные скрипты ARIS

### 10.3.1. Общие сведения о файлах скриптов

Стандартные скрипты модуля *ARIS Report* располагаются в каталоге *ARIS5\Script\Report\en*. Они различаются по назначению в соответствии с классом элементов, для обработки которых они предназначены. Различия отражаются в третьей (последней) букве расширения файла скрипта (см. табл. 50).

Таблица 50. Расширения файлов скриптов

Назначение	Расширение файла
Отчет общего назначения	*rsc
Отчет по объектам базы данных	*rso
Отчет по моделям базы данных	*rsm
Отчет по структуре базы данных (папках базы данных)	*rsg
Отчет по фильтрам базы данных	.rsf
Отчет по базам данных сервера	.rsd
Отчет по таблицам базы данных	.rst

Сгенерированные отчеты по умолчанию располагаются в каталоге *ARIS5\Script\Report\out*.

10.3.2. Описание скриптов для отчетов по базам данных

*BSC\_DataImport.rsd*. Осуществляет импорт данных (из Excel в ARIS), относящихся к методике оценки стратегии компании Balanced Score Card, после того как они были перенесены туда скриптом *BSC\_DataExport.rsm* и изменены в соответствии с текущим состоянием стратегии компании.

*DBLanguagesFontFormats.rsd*. Выводит информацию о всех используемых в базе данных языках и форматах шрифтов.

*DBUser.rsd*. Скрипт описывает всех пользователей и их группы для выбранной базы данных (см. раздел 9.8.1) и выдает информацию относительно назначенных для них функциональных привилегий. Пример отчета, выполненного по стандартному скрипту *DBUser.rsd*, приведен на рис. 208. Использование английского языка объясняется тем, что отчет выполнен с использованием нелокализованной версии ARIS.

*IDCheck.rsd*. Осуществляет проверку идентификаторов всех элементов базы данных на наличие в них символов, зависящих от используемого языка. При необходимости можно выбрать опцию удаления этих символов.

*Translation\_In.rsd*. Служит для повышения удобства перевода содержимого баз данных с одного языка на другой. Осуществляет

перенос в ARIS информации, предварительно переведенной в системе Excel (см. также скрипт *Translation\_Out.rsg*).

10.3.3. Описание скриптов для отчетов по папкам базы данных

*CMA\_Activities.rsg*. Формирует список действий по планируемым изменениям в моделях и объектах (см. раздел 5.8, модуль *ARIS Change Management*), о которых должен быть проинформирован разработчик моделей или менеджер проекта.

*CMA\_improvements.rsg*. Генерирует в формате XLS таблицу объектов и моделей, для которых имеются предложения по усовершенствованию.

*CMA\_StatusChanges.rsg*. Генерирует список объектов и моделей, состояние которых изменилось в рамках работ по управлению изменениями (модуль *ARIS Change Management*).

	Function rights of database users/user groups of the database	Весть - МетаТехнология
---	---	------------------------

## ARIS Report

Server: LOCAL

Database: Методология ARIS

User: system

User group	User	Show user management	User management	Change management	Database management	Database export	Method changes	Prefix management	Font format management
ARIS Seminar				X			X		X
	VictorK			X			X		X
	OlegZ			X			X		X
ARIS Seminar		X	X	X	X	X	X	X	X
	system	X	X	X	X	X	X	X	X
	AntonS	X	X	X	X	X	X	X	X
ARIS Seminar				X	X	X	X	X	X
	IvanT			X	X		X	X	X
	NikolayC			X	X	X	X	X	X

Рис. 208. Отчет, полученный с использованием скрипта DBUser.rsd

*Group Info.rsg.* Выводит структуру папок.

*GroupModels.rsg.* Формирует информацию о всех моделях выбранной папки и вложенных в нее папок. Имя модели и ее тип выводятся по умолчанию. Имеется возможность включения выбранных атрибутов в отчет. Можно также импортировать в отчет графику моделей (для форматов HTML, RTF, DOC).

*GroupModels\_r3.rsg.* Выводит информацию о всех объектах типа *Функция*, которые содержатся в рассматриваемых моделях и удовлетворяют критериям выбора. Критерий выбора задается перед созданием отчета и

позволяет ограничить число объектов, включаемых в отчет.

*GroupObjects.rsg.* Собирает информацию об объектах, содержащихся в выбранной папке.

*Translation \_Out.rsg.* Служит для повышения удобства перевода содержимого баз данных с одного языка на другой. Осуществляет перенос информации из ARIS в Excel. Данная операция необходима для перевода служебной информации и занесения ее обратно в ARIS при помощи скрипта *Translation \_Out.rsg.*

#### 10.3.4. Описание скриптов для отчетов по моделям

*Activation\_R3.rsm.* Выводит информацию о состоянии активации объектов заданных типов — функций, события и системных организационных единиц. Обработывается информация об имени, типе объекта, его отнесении к критерию активности, а также код транзакции в информационной системе. Применяется для моделей, имеющих отношение к системе управления предприятием SAP R/3, в частности, моделей eEPC, VAD и матрицы выбора процессов.

*Assigmnents\_R3.rsm.* По каждой модели, имеющей отношение к системе SAP R/3, выводит перечень функций, для которых имеется детализация. Обработывается информация о коде транзакции, абсолютном пути к детализирующей модели, типе детализирующей модели, символе анализируемого объекта и статусе его активизации. Используемые модели те же, что и в предыдущем случае.

*Aris2xmiRR.rsm (Aris2xiniRR\_v2.rsm).* Экспортирует все выбранные модели типа UML-диаграммы классов в файл формата XML в нотации UML XMI. Полученная информация может быть импортирована в программный продукт Rational Rose 2000 (или любой другой продукт, поддерживающий стандарт XMI). Для каждой выбранной модели создается отдельный экспортный файл.

*BSC\_DataExport.rsm.* Осуществляет экспорт данных из ARIS в Excel, относящихся к методике оценки стратегии компании Balanced Score Card. Эти данные могут быть изменены в Excel и перенесены обратно при помощи скрипта *BSC\_DataImport.rsd*. Применяется для моделей: причинно-следственная диаграмма BSC, дерево

функций, организационная схема, модель структуры и диаграмма VAD.

*BSC\_Output.rsm.* Выводит для всех диаграмм типа «причинно-следственная диаграмма BSC», связанных с моделями организационных схем, моделями структуры, деревьями функций и диаграммами VAD, следующую информацию: стратегические цели, ключевые показатели результативности, задачи. Эта информация предоставляется в различных комбинациях по выбору пользователя. Применяется для тех же моделей, что и в предыдущем случае.

*CBusinessScenarioAggregated.rsm.* Выводит общую информацию об участниках бизнеса и их деловых процессах с отображением входов и выходов. Отчет создается для моделей типа «Диаграмма сценариев e-бизнеса».

*CBusinessScenarioDetailed.rsm.* Выводит детальную информацию об участниках бизнеса и их деловых процессах с отображением входов и выходов рассматриваемых процессов. Отчет создается для моделей типа «Диаграмма сценариев e-бизнеса».

*Function\_r3.rsm.* Используется для обзора функций, имеющих отношение к системе SAP R/3 (модели eEPC, VAD и матрица выбора процессов). Содержит информацию о коде транзакции, критерии классификации, состоянии активации. Возможен вывод графики модели.

*Function.Selection\_r3.rsm.* Используется для обзора функций, имеющих отношение к системе управления SAP R/3 и удовлетворяющих заданным критериям, выбираемым перед генерацией отчета. Содержит информацию о коде транзакции, критерии классификации, состоянии активации. Возможен вывод графики модели. Применяется для тех же моделей, что и в предыдущем случае.

*Model\_3x.rsm.* Осуществляет вывод свойств и атрибутов выбранных моделей и содержащихся в них объектов. Состав выводимой информации определяется пользователем в ходе диалога. Отчет создается в текстовом формате.

*ModelGraph.rsm.* Выводит в текстовом формате информацию об объектах выделенных моделей.

*ModelGraphics.rsm.* Предназначен для переноса графики модели из системы ARIS в офисные приложения. Пользователем задается формат вывода — ориентация листа, его размер и поля, правила разделения модели на страницы, коэффициент масштабирования (задаваемый пользователем, оптимальный или соответствующий масштабу для печати).

*ModelHierarchy.rsm.* Осуществляет обработку функций и их детализаций, содержащихся во всех выбранных моделях. Формируется иерархия функций с выводом свойств и атрибутов. Глубина представляемой иерархии задается пользователем. Функции модели могут выводиться в алфавитном порядке, сортироваться по любым атрибутам, типам символа или топологии. Информация о функциях, имеющих множественные вхождения, выводится один раз. Все дополнительные экземпляры отображаются в виде ссылок.

*ModelHierarchy\_2.rsm.* Осуществляет обработку объектов и их детализаций, содержащихся во всех выбранных моделях. В отчете формируется иерархия функций с выводом их свойств и атрибутов.

*Modellnfo.rsm.* Позволяет получить описание всех существующих атрибутов для выбранных моделей. По желанию в отчете можно указать папку, в которой модели сохранены, вывести графику модели и объектов с их атрибутами. Кроме того, могут выводиться связи между объектами в пределах модели. Набор выводимых типов моделей, объектов, отношений и атрибутов может быть ограничен выбираемым фильтром обработки. Содержание отчета сортируется в

алфавитном порядке. Вывод может производиться в форме таблицы или текстового документа.

*Modellso.rsm.* Выводит информацию с учетом требований международных стандартов ИСО серии 9000. Выводятся элементы, группы, установленные атрибуты и изображения моделей. Функции, имеющиеся в моделях, могут выводиться в алфавитном порядке, сортироваться по любому атрибуту или типу символа. Атрибуты, отношения и детализирующие модели сравниваются. Вывод информации производится в текстовом формате.

*ModelObjects.rsm.* Включает в отчет все объекты выбранных моделей. При составлении отчета рассматриваются только те типы объектов, которые входят в установленный (стандартный) методологический фильтр. Отчет генерируется в виде таблиц (для каждой модели), объекты располагаются в алфавитном порядке.

*ModelObjectTable.rsm.* Выводит для объектов выбранной модели заданные атрибуты. Вывод производится в табличном формате.

*ModelVariants.rsm.* Выводит в отчет результаты сравнения выбранных моделей и их вариантов. Сравнение проводится до уровня, устанавливаемого пользователем. Вывод производится в текстовом формате, модели располагаются в алфавитном порядке.

*TreeNumbering.rsm.* Подсчитывает число объектов выбранной модели с учетом иерархии и выводит результаты на изображение модели.

*OrgChat\_1.rsm.* Описывает структуру для выбранных организационных схем. Все разрешенные в методологическом фильтре типы связей будут включены в отчет. В опциях скрипта можно задать способ представления объектов — с учетом или без учета иерархических отношений. Отчет выводится в текстовом формате.

*ProcessOverview\_1.rsm*, *ProcessOverview\_2.rsm*. В отчете описываются функции, входящие в выбранные модели типов eEPC, eEPC (с потоком материалов), а также в офисный и производственный процессы. Могут быть также учтены включенные в эти модели организационные единицы, связанные с функциями. Анализируются все возможные типы связей между организационными единицами и функциями. Кроме того, объекты, описывающие данные, могут быть приведены в отчете как входные и выходные данные функций. Графическое изображение модели также может быть включено в отчет (за исключением использования выходного текстового формата). Объекты сводятся в таблицу и располагаются в алфавитном порядке. Различие между этими скриптами состоит в типах учитываемых связей между объектами.

*ProcessOverview\_3.rsm*. В отчете описываются функции, входящие в выбранные модели типов eEPC, eEPC (с потоком материалов), а также в офисный и производственный процессы. Также могут быть учтены включенные в эти модели организационные единицы, связанные с функциями. В зависимости от опции, установленной при генерации отчета, анализ может проводиться в обратном направлении, т.е. рассматриваются функции и выполняющие их организационные элементы. В зависимости от выбранной опции отчет может быть представлен в табличной или текстовой форме.

*ProcessOverview\_4.rsm*. Предназначен для моделей типов eEPC, eEPC (с потоком материалов), офисного и производственного процессов. Генерирует отчет, который включает организационные единицы и выполняемые ими функции. Окружение функций анализируется для типов связи *выполняет (executes)* или *выполняется (is executed by)*. В зависимости от выбранной опции отчет представляется в виде таблицы или текста.

*ProcessOverview\_5.rsm*. Предназначен для моделей типов eEPC, eEPC (с потоком

материалов), офисного и производственного процессов. Отчет описывает данные ввода и вывода, представленные объектами типа *Кластер/модель данных. Тип сущности. Технический термин* и *Информационный носитель*, а также функции, с которыми они связаны. Анализируются типы связи *является входом для (is input for)* и *имеет на выходе (has output of)*. При выводе в отчет объекты сортируются в алфавитном порядке.

*ProcessOverview\_6.rsm*. Предназначен для моделей типов eEPC, eEPC (с потоком материалов), офисного и производственного процессов. Отчет описывает объекты, характеризующие тип прикладной системы, и функции, которые они поддерживают. В зависимости от установленных опций отчет может быть представлен в обратном направлении, т.е. покажет связи функций модели eEPC с объектами, описывающими типы прикладной системы. В отчете анализируется тип связи *может поддерживать (can support)* или *может поддерживаться (can be supported by)*. Объекты располагаются в алфавитном порядке в формате таблицы или как текст.

*ProcessOverview\_7.rsm*. Предназначен для моделей типов eEPC, eEPC (с потоком материалов), офисного и производственного процессов. Отчет описывает функции, имеющие среди своих заполненных атрибутов временные характеристики и/или параметры стоимости/затрат. В зависимости от настройки опций выбираются для включения в отчет средние, минимальные или максимальные значения. Суммарные характеристики выводятся в конце отчета. Для подведения итогов по временным атрибутам различные единицы измерения могут быть сведены к одной из них. Для подведения итогов по затратам одна и та же валюта должна быть использована для различных стоимостных атрибутов. Сумма затрат в функции включает все типы стоимости, кроме общих стоимостей. В отчет не включаются результаты анализа различных путей процесса. Функции выводятся в алфавитном порядке в виде таблицы

*ProcessPath\_r3.rsm.* Показывает связи между интерфейсами процессов.

*StructuralModel.rsm.* Описывает модели, содержащие структурно-зависимые объекты. Информация выводится в формате таблицы или как текст.

*VACD\_1.rsm.* Предназначен для диаграммы цепочек добавленного качества. Отчет описывает атрибуты и графические символы модели (за исключением использования текстового выходного формата). В отчет включаются также функции с атрибутами *название* и *описание/определение*. В зависимости от установленных опций в отчете могут быть представлены все организационные элементы вместе с выполняемыми функциями и данными ввода и вывода. Для объектов, описывающих данные, анализируются типы связи *является входом для (is input for)* и *имеет на выходе (has output of)*. Для организационных элементов рассматриваются все возможные типы связи между этими элементами и функциями. Отчет составляется в алфавитном порядке и выводится в виде таблицы.

*VACD\_2.rsm.* Этот скрипт аналогичен предыдущему, однако анализируются следующие типы связи: функция *выполняется (is executed by)* организационным элементом;

объект, описывающий данные, *является входом для (is input for)* функции; функция *имеет на выходе (has output of)* объект, описывающий выходные данные. Отчет составляется в алфавитном порядке в формате таблицы.

### 10.3.5. Описание скриптов для отчетов по таблицам функционально-стоимостного анализа

*CostCenterAnalysis\_1.rst.* Выводит результаты анализа затратных центров, выполненного методом ABC.

*ProcessCalculation.rst.* Выводит результаты расчета стоимости процессов, произведенного методом ABC.

### 10.3.6. Описание скриптов для отчетов по объектам

*FunctionalHierarchy.rso.* Выводит в отчет иерархию функций в базе данных. Для выбранного объекта типа *функция* фиксируются все связанные с ней функции, содержащиеся в базе данных. При формировании отчета анализируется тип связи *подчиняется по процессу (is a process-oriented subordinate)*. Функции, относящиеся к одному уровню иерархий, сортируются в алфавитном порядке. Отчет выводится в текстовом формате.

*Functions\_1.rso.* Описывает взаимосвязь выбранных функций и организационных единиц. Анализируются все возможные связи между объектами типа *функция* и организационными единицами. При формировании отчета функции располагаются в алфавитном порядке. В зависимости от выбранных опций отчет выводится в виде таблицы или в текстовом формате.

## Документирование моделей

*Functions\_2.rso*. Описывает организационные элементы, выполняющие выбранные функции. Анализируется только тип связи *выполняется (is executed by)* между организационным элементом и функцией. Объекты при выводе в отчет сортируются в алфавитном порядке. В зависимости от выбранных опций отчет представляется в виде таблицы или в текстовом формате. Пример отчета, выполненного с

использованием данного скрипта и содержащего информацию о том, какая функция выполняется каким организационным подразделением, приведен на рис. 209.

*Object\_3x.rso*. Генерирует отчет, содержащий информацию о папках, моделях и находящихся в них выделенных объектах и их связях.

	Function and their executing organizational items		
<b>ARIS Report</b>			
Server: LOCAL			
Database: Методология ARIS			
User: system			
Function	Relationship type	Object name	Object type
Идентифицировать изделие	No corresponding types		
Идентифицировать неисправимый брак	No corresponding types		
Идентифицировать готовые изделия	is executed by	Отдел контроля	Organizational unit
Исправлять несоответствие параметров	No corresponding types		
Определять параметр, не соответствующий норме	No corresponding types		
Отправлять на склад	No corresponding types		
Проверить параметры А, В и С	is executed by	Отдел контроля	Organizational unit
Провести утилизацию изделия	is executed by	Производственный отдел	Organizational unit

Рис. 209. Отчет для модели eEPC, полученный с использованием скрипта *Functions\_2.rso*

*ObjectInfo.rso*. Предоставляет для выбранных объектов информацию о всех их связях с другими объектами. Установленные атрибуты анализируемых объектов могут быть включены в отчет, который имеет форму таблицы.

*ObjectTables.rso*. Формирует отчет о всех установленных атрибутах выбранных объектов и их связях с другими объектами.

*ObjectVariants.rso*. Выводит информацию о результатах сравнения выбранных объектов и их типов. Объекты и их типы на структурном уровне располагаются в алфавитном порядке. Отчет имеет форму таблицы.

*Occurrences.rso*. Выводит в отчет информацию о местонахождении модели в базе данных ARIS.

*OrgUnits\_1.rso*. Описывает для выбранных организационных единиц все относящиеся к ним функции. В отчет выводятся существующие типы связи между организационными единицами и функциями. Кроме того, в отчет могут быть выведены атрибуты организационных единиц. В зависимости от опций отчет представляется в виде таблицы или как текстовый документ. Объекты сортируются по их типам, в рамках одного типа — в алфавитном порядке.

*OrgUnits\_2.rso*. Описывает функции, которые выполняют выбранные организационные единицы. Анализируется тип отношений *выполняет* (*executes*). В отчет могут быть включены атрибуты организационных единиц. В зависимости от выбранных опций отчет представляется в виде таблицы или как текстовый документ. Объекты сортируются по их типам, а в рамках одного типа — в алфавитном порядке.

## Моделирование бизнеса. Методология ARIS

*OrgUnits\_3.rso*. Выводит в отчет связи всех выбранных организационных единиц с другими организационными единицами. Анализируются все типы связей, которые разрешаются между организационными единицами в установленном методологическом фильтре. Атрибуты выбранных объектов и папка, в которой они находятся, могут также включаться в отчет. Отчет представляется как в текстовом формате, так и в виде таблицы.

### 10.3.7. Описание скриптов для отчетов по фильтрам

В ARIS имеется только один файл скрипта для фильтров — *FilterInfo.rsf*. Он выводит информацию о входящих в установленный методологический фильтр моделях, объектах, их атрибутах, разрешенных связях и т.д. Пользователь может выбирать нужную ему информацию (рис. 210). Вывод осуществляется в виде таблицы. Отчет позволяет получить полную информацию о любой модели ARIS.

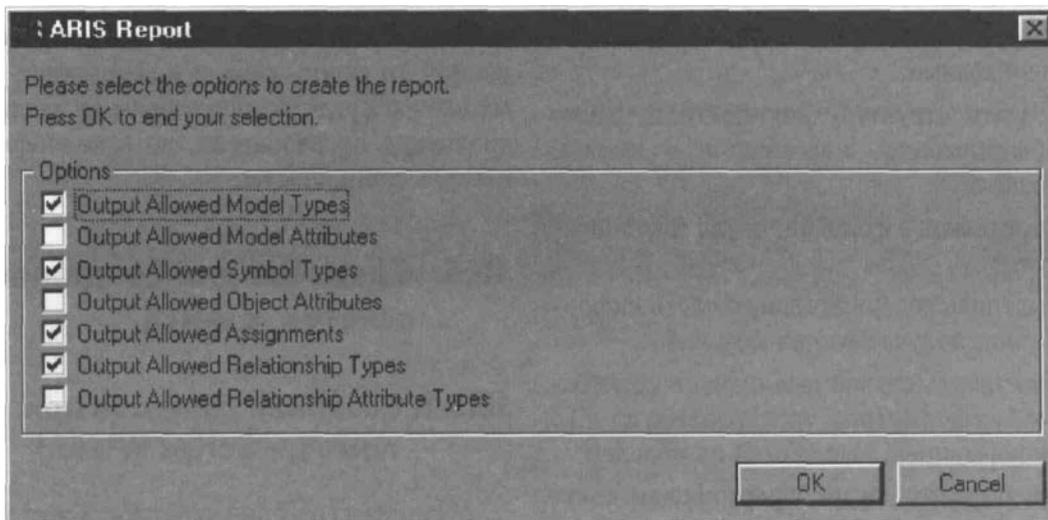


Рис. 210. Настройка скрипта для отчета по фильтрам

#### 10.4. Рекомендации по документированию моделей

Структура и состав выводимой в отчет информации могут быть определены пользователем. Для этого необходимо разработать собственный скрипт при помощи специального модуля *ARIS Script Editor* — редактора скриптов. Эта возможность является очень важным свойством ARIS.

Однако получение скрипта, позволяющего генерировать качественные отчеты, связано не только с его программированием. В первую очередь следует продумать содержание, структуру и форму выводимой информации. Таким образом, в процессе создания скриптов можно выделить ряд этапов:

- определение целей отчета;
- определение групп пользователей;
- определение требований к отчету в соответствии с запросами групп пользователей;
- оценка возможностей ARIS по выводу информации в нужном виде и изменение формы отчетов в соответствии с требованиями ARIS;

- оценка и корректировка (при необходимости) требований к моделированию;
- создание скриптов отчетов с помощью механизма *Script Wizard (Помощник создания отчетов)*;
- корректировка и русификация (при использовании английской версии системы ARIS) скриптов отчетов в модуле *ARIS Script Editor*;
- генерация отчетов.

При подготовке формы отчета (технического задания на разработку скрипта) необходимо определиться с выводимой информацией, а также выделить элементы репозитория ARIS, которые содержат интересующую информацию, в частности, информацию о документируемых моделях, объектах, которые будут выводиться в документ, их характеристиках и атрибутах, характеристиках отношений между объектами, а также информацию о детализирующих объектах.

Важным этапом является определение форм представления данных — структурированный текст, таблицы, графика модели.

После решения перечисленных выше вопросов необходимо:

- создать структуру документа и форму представления информации в каждом разделе;
- определить заголовки, стили и колонтитулы;
- выделить содержательную часть информации, получаемую из моделей;
- расставить служебные слова и согласовать их форму (род, число, падеж, и т.д.) с информацией, выводимой из моделей.

Таким образом, будет сформирован документ с требованиями к написанию скрипта. После этого можно переходить непосредственно к программированию.

В ARIS можно использовать специальный механизм *Script Wizard* — визуальное средство, позволяющее создать структуру (шаблон) скрипта, т.е. определить, какие модели, объекты, связи и атрибуты будут обрабатываться этим скриптом и какие элементарные формы фильтрации и сортировки целесообразны при этом. Благодаря механизму *Script Wizard* можно быстро получить набросок желаемой

структуры вывода информации, поэтому данное средство применяют не только начинающие пользователи, но и квалифицированные специалисты.

## 10.5. Средства для создания отчетов в ARIS

### 10.5.1. Создание скриптов при помощи Script Wizard

Механизм создания скриптов *Script Wizard* позволяет также быстро построить новую программу для генерации отчета. Он реализуется посредством диалоговых окон, перемещаясь по которым пользователь формирует структуру будущей программы.

Для запуска *Script Wizard* необходимо выбрать пункт меню *File/New*, после чего появится диалоговое окно (рис. 211). При выборе пункта меню *Evaluation Script* открывается первое окно *Script Wizard* (рис. 212).

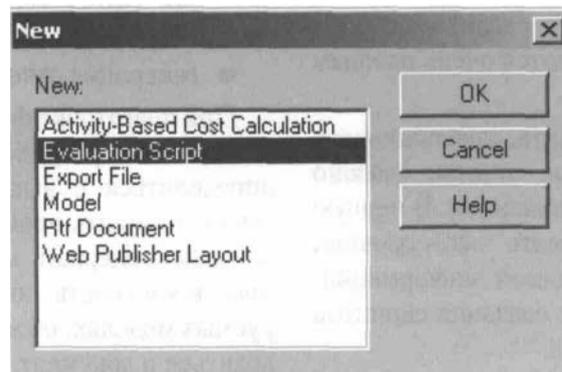


Рис. 211. Выбор типа документа

## Документирование моделей

В нем определяются основные параметры будущего скрипта. В первом списке (поле *For which component do you wish to create the script?*) необходимо указать тип скрипта *Report (Отчет)*. Во втором списке (поле *For which context do you wish to create the script?*) указывается назначение скрипта — для объектов, моделей, папок и т. д. В текстовом поле *Script file name* необходимо указать имя

файла, в котором будет сохранен создаваемый скрипт. В области *Generation (Генерация)* необходимо выбрать опцию *Generate Code (Генерировать код)*, которая осуществляет переход к дальнейшим шагам мастера создания скрипта. Если выбрать опцию *Create Empty Script (Создать пустой скрипт)*, то система создаст программу скрипта, содержащую только заголовки.

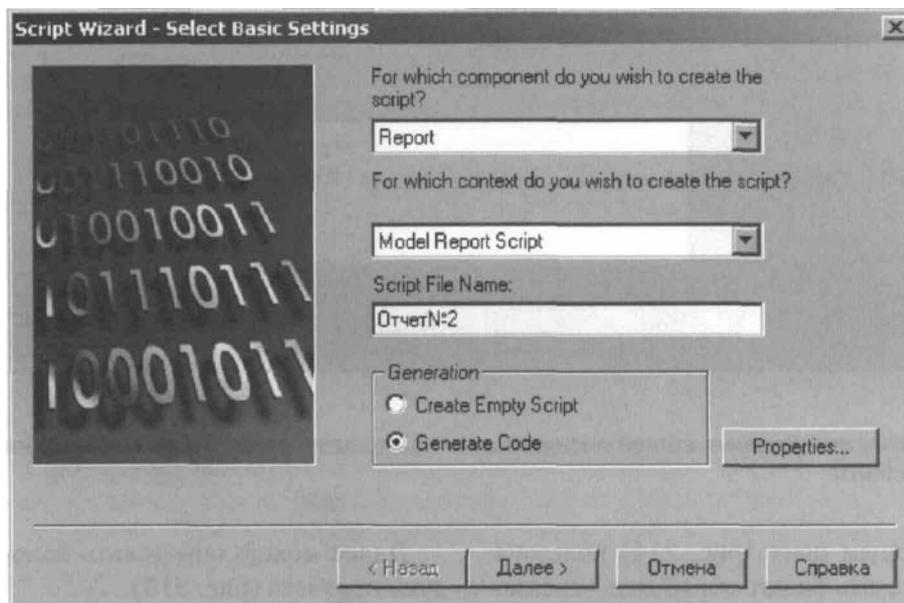
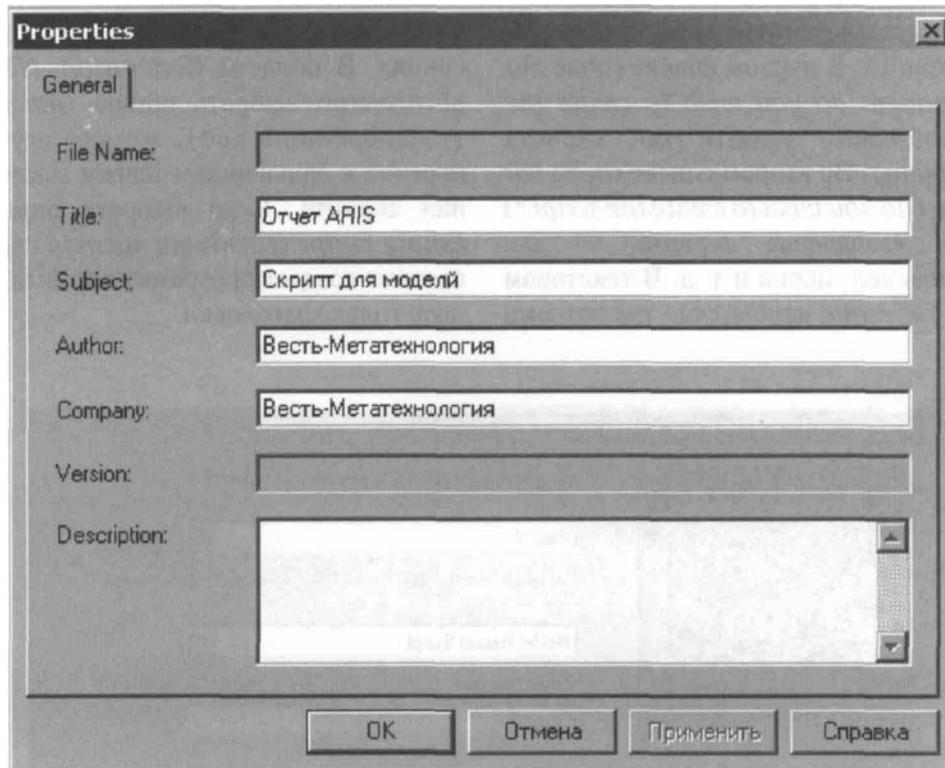


Рис. 212. Окно Script Wizard: указание основных параметров отчета

В этом же окне можно определить такие параметры будущего отчета, как заголовок, автор, название компании, краткое описание

скрипта и другие. Эти параметры задаются в диалоговом окне, вызываемом нажатием кнопки *Properties (Свойства)* (рис. 213).



**Рис. 213. Окно для указания общей информации — заголовок, автор, название компании, описание скрипта**

На следующем шаге (рис. 214) пользователь может указать фильтр обработки, через который скрипт будет «видеть» содержимое базы данных. Кроме этого, пользователь может указать, для каких типов объектов и моделей рекомендуется применять создаваемый скрипт.

Далее можно определить колонтитулы будущего отчета (рис. 215).

Столбцы таблицы в колонтитулах соответствуют полям диалога. На рис. 216 показано, как выглядит верхний и нижний колонтитулы отчета.

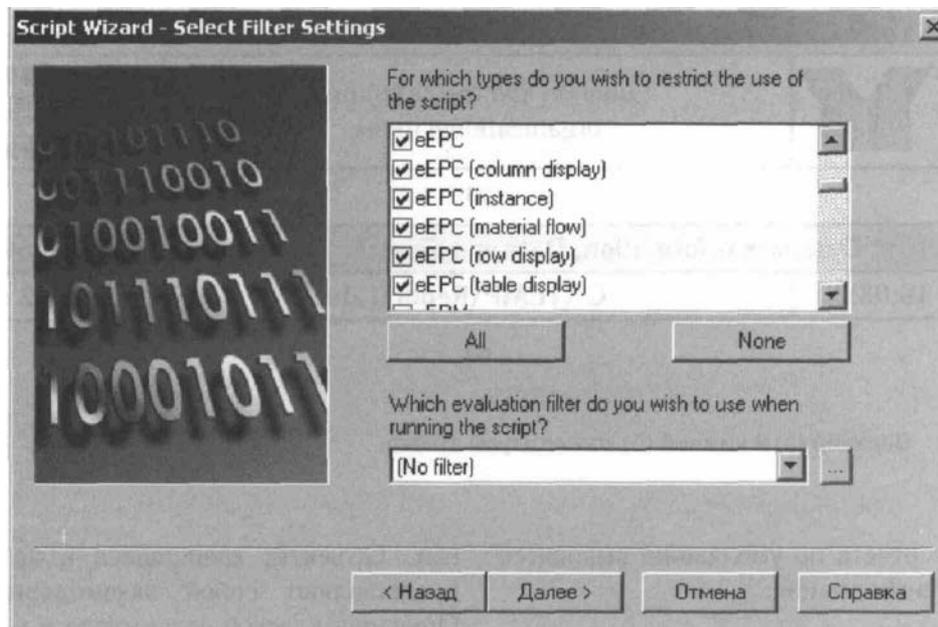


Рис. 214. Окно Script Wizard: выбор фильтра и ограничений для применения скрипта

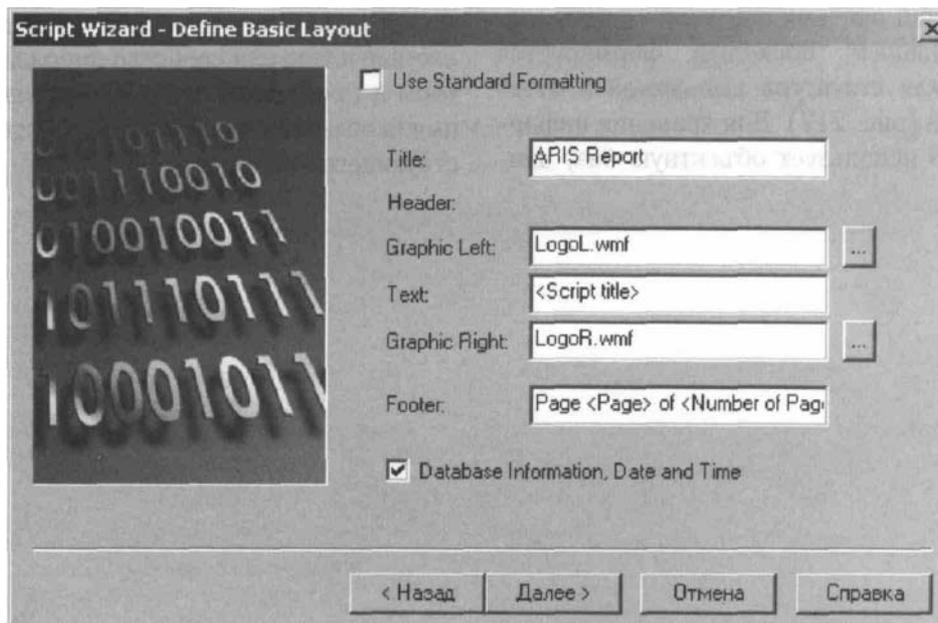


Рис. 215. Окно Script Wizard: задание элементов для оформления отчета

Graphics Left	Text	Graphics Left
	Function and their executing organizational items	

a)

Database Information, Date and Time	Footer
17.05.00 16:03:23	Page 2 of 10

б)

Рис. 216. Верхний (а) и нижний (б) колонтитулы отчета

В начале отчета по умолчанию выводится следующая информация:

- заголовок;
- название сервера;
- название базы данных;
- имя пользователя.

Следующий шаг при создании скрипта является основным, поскольку формируется иерархическая структура выводимой в отчет информации (рис. 217). Для хранения информации ARIS использует объектную базу данных. Объекты,

хранящиеся в базе данных, представляют собой экземпляры классов. Описание классов содержится в контекстной справке по системе ARIS (раздел *ARIS Script Editor*).

В правом диалоговом окне пользователь видит список классов и их свойств, которые он может добавить в программу. Каждый добавляемый класс или свойство порождают операторы в программе, которые реализуют вывод имени объекта или значения свойства соответствующего класса.

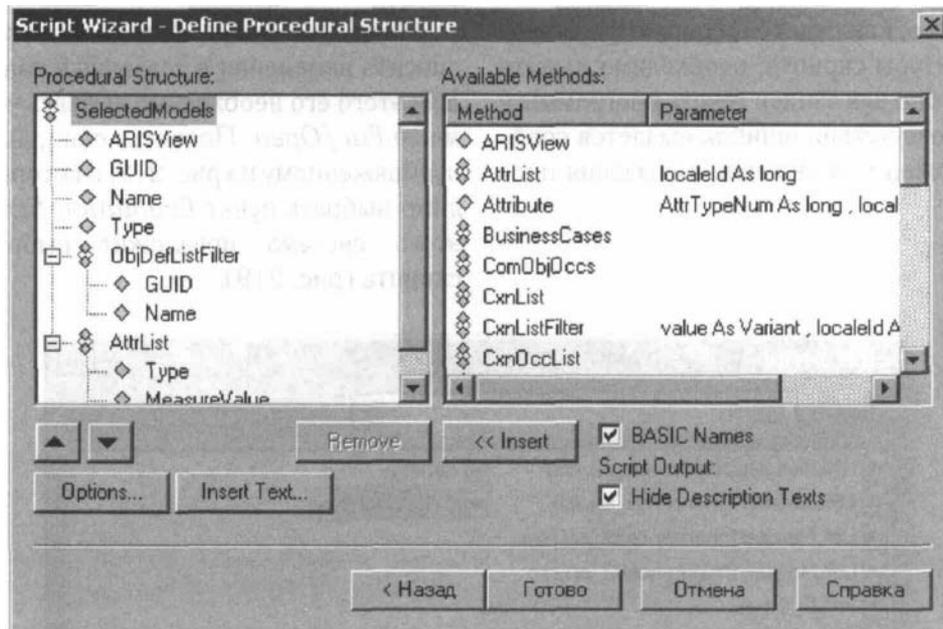


Рис. 217. Окно Script Wizard: формирование иерархической структуры отчета

Добавляемый класс или свойство может являться списком. Списки помечены символом, состоящим из двух ромбиков. В этом случае в отчет выводится содержимое списка. Пользователь может упорядочить

список, указав до трех критериев его сортировки. При нажатии *Options (Опции)* появится диалоговое окно сортировки списка (рис. 218).

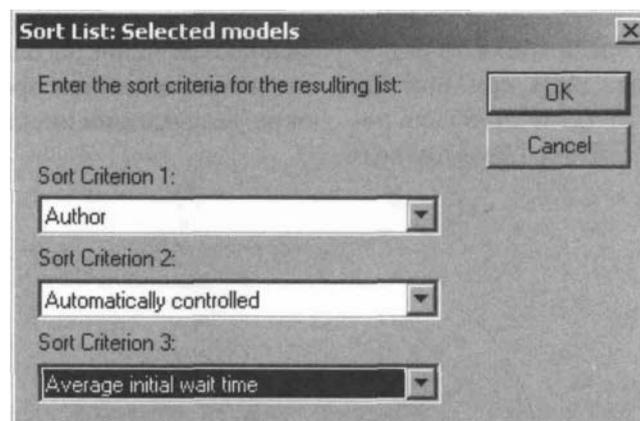


Рис. 218. Диалоговое окно сортировки списка

### Моделирование бизнеса. Методология ARIS

После того, как будет завершено формирование структуры скрипта, необходимо нажать кнопку *Готово* для записи текста программы в файл. При отсутствии ошибок выдается сообщение об успешном окончании создания программы.

Механизм *Script Wizard* позволяет также вносить изменения в созданный ранее скрипт. Для этого его необходимо вызвать через пункт меню *File/Open*. Появится окно, аналогичное изображенному на рис. 212, в котором необходимо выбрать пункт *Evaluation Script*. После этого система предложит выбрать файл скрипта (рис. 219).

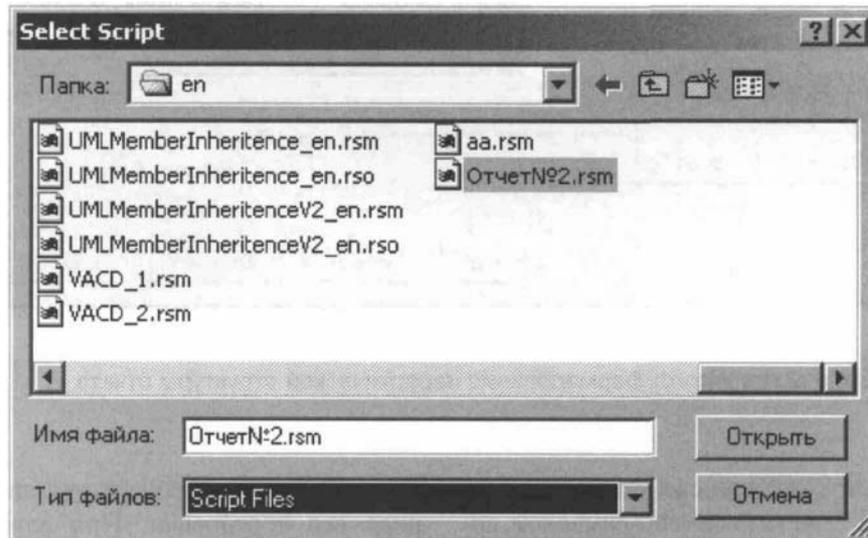


Рис. 219. Выбор файла скрипта для внесения изменений

Выбрав файл, следует задать способ редактирования скрипта (рис. 220) при помощи *Script Wizard* (опция *In Script Wizard*) или редактора (опция *In Script Editor*). При

этом возможно изменение заголовка, автора, названия компании, описания программы в диалоговом окне, вызываемом кнопкой *Properties*.

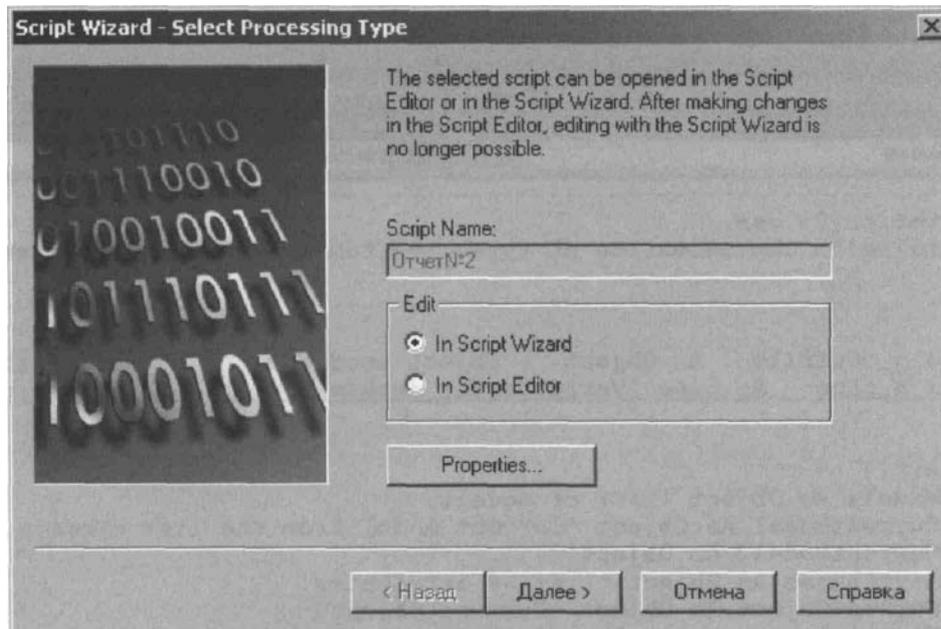


Рис. 220. Окно Script Wizard: выбор способа редактирования скрипта

При выборе опции *In Script Wizard* повторяется последовательность действий, описанных начиная с рис. 214.

Если изменения, которые нужно внести, выходят за рамки возможностей механизма *Script Wizard* (например, требуется более сложная процедура выборки данных, чем фильтрация по набору атрибутов), то следует открыть скрипт в модуле *ARIS Script Editor* (опция *In Script Editor*). Краткое описание этого модуля приведено в разделе 10.5.2.

При использовании *Script Wizard* важно помнить о следующем ограничении: если разрабатываемый скрипт был открыт и сохранен при помощи модуля *ARIS Script Editor*, то этот скрипт уже нельзя будет открыть при помощи механизма *Script Wizard*.

### 10.5.2. Создание скриптов в модуле ARIS Script Editor

Модуль *ARIS Script Editor* входит в *ARIS Toolset* и дает возможность пользователю при необходимости создавать и редактировать собственные скрипты отчетов в дополнение к скриптам, включенным в инструментальную систему ARIS.

Для того, чтобы создать новый скрипт, нужно выбрать пункт меню *File/New/Evaluation Script*. Для редактирования существующего скрипта выбирается пункт меню *File/Open/ Evaluation Script*.

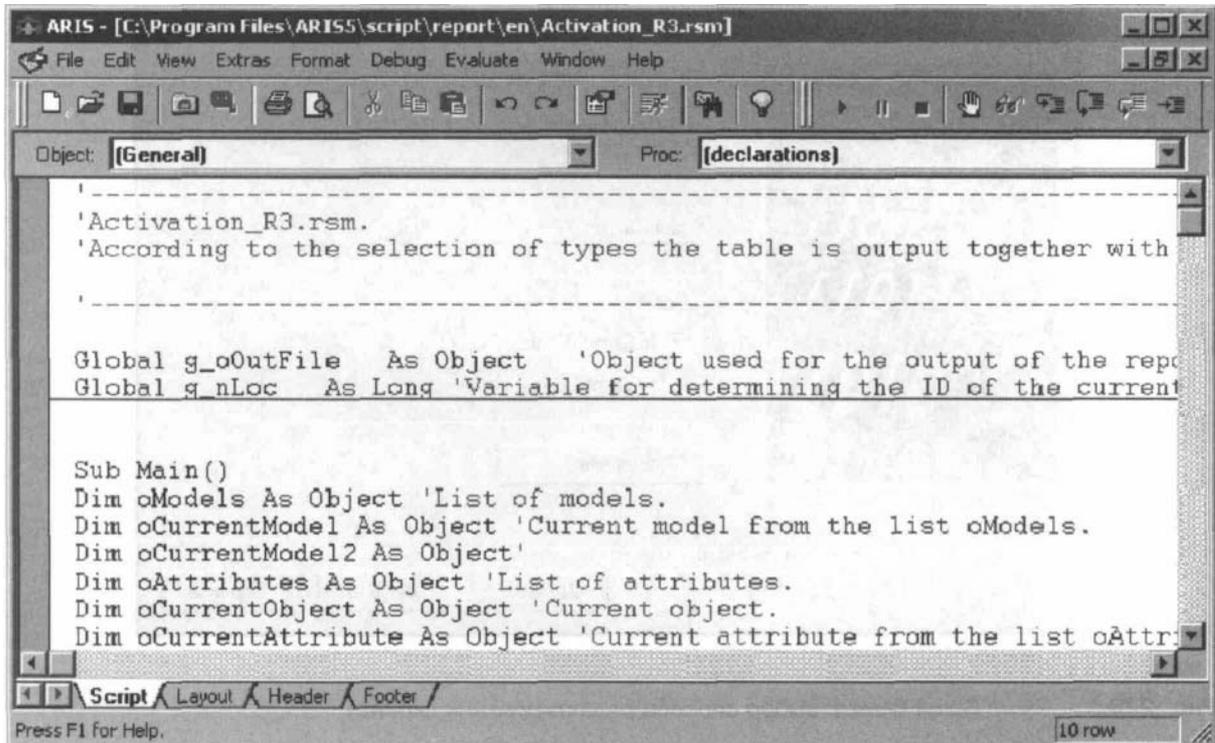


Рис. 221. Экрана редактора скриптов ARIS Script Editor

Таблица 51. Семантическая раскраска программы

Семантический смысл	Цвет
Комментарии	Зеленый
Команды языка APIS ©Basic Language Capability	Темно-синий
Команды Visual Basic®	Синий
Структуры языка (определенные пользователем объекты, методы этих объектов, константы и т.д.)	Черный
Строки, содержащие синтаксические ошибки	Красный
Определения контрольных терминов	Бирюзовый

Экран редактора скриптов *ARIS Script Editor* приведен на рис. 212. В заголовке окна видно полное имя файла, в котором хранится программа. Ниже располагаются главное меню системы и панель инструментов. Далее находится окно модуля, в верхней части которого — два раскрывающихся списка. Правый список отражает структуру программы (список всех процедур и функций, блок глобальных объявлений переменных) и позволяет перемещаться между структурными блоками программы.

Редактор скриптов выделяет цветом семантически различные участки кода программы. Применяемые для этого цвета приведены в табл. 51.

В нижней части окна имеются закладки для просмотра текста программы *Script* (по умолчанию активно), макета отчета *Layout* (в режиме «только чтение»), верхнего (*Header*) и нижнего (*Footer*) колонтитулов.

Модуль *ARIS Script Editor* имеет встроенную систему просмотра библиотек, используемых для создания скриптов. Данная система позволяет найти и вставить в скрипт интересующий пользователя метод или свойство, а также просмотреть формат задания параметров.

С помощью *ARIS Script Editor* можно генерировать отчеты любой сложности, соответствующей возможностям алгоритмического языка программирования и существующих наборов методов и свойств библиотек ARIS, которые обеспечивают доступ к базе данных.

# 11. Глоссарий ARIS: модели, входящие в методологические фильтры, типы объектов и типы связей

## Организационные модели

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP-фильтр (SAP Filter)	Фильтр для модуля ARIS for MY SAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Организационные модели. Уровень определения требований</b>							
1	Организационная схема Organizational chart							
2	Календарь смен Shift calendar							
	<b>Организационные модели. Уровень спецификации проекта</b>							
1	Топология сети Network topology							
	<b>Организационные модели. Уровень описания реализации</b>							
1	Диаграмма сети Network diagram							
2	Технические ресурсы Technical resources							

## Функциональные модели

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Extended Standard Filter - Расширенный фильтр	SAP-фильтр SAP Filter	Фильтр для модуля ARIS for MySAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Функциональные модели. Уровень определения требований</b>							
1	Дерево функций Function tree							
2	Диаграмма целей Objective diagram							
3	SAP ALE функциональная модель SAP ALE function model							
4	SAP Диаграмма приложений SAP applications diagram							
5	Y-диаграмма Y-diagram							
	<b>Функциональные модели. Уровень спецификации проекта</b>							
1	Диаграмма типа прикладной системы Application system type diagram							
	<b>Функциональные модели. Уровень описания реализации</b>							
1	Диаграмма прикладной системы Application system diagram							

Моделирование бизнеса. Методология ARIS

Модели данных

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP-фильтр (SAP Filter)	Фильтр для модуля ARIS for MySAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Модели данных. Уровень определения требований</b>							
1	Иерархия полномочий Authorization hierarchy							
2	Диаграмма стоимостных факторов CD Diagram						■	
3	Диаграмма категории затрат Ccost category diagram						■	
4	Диаграмма структуры хранилища данных Data Warehouse (DW) - Structure diagram							■
5	Расширенная модель "сущность - отношение" eERM eERM (Extended entity relationship model)		■	■		■		■
6	Диаграмма атрибутов модели eERM eERM attribute allocation diagram		■	■				■
7	Модель данных IEF IEF Data Model							
8	Диаграмма информационных носителей Information carrier diagram							
9	Диаграмма структуры знаний Knowledge structure diagram			■				
10	Диаграмма материалов Material diagram							
11	SAP SERM - структурированная модель "сущность-отношение" SAP SERM (SAP structured entity relationship model)				■			
12	Модель семантики данных SeDaM (Semantic data model)							
13	Модель технических терминов Technical terms model	■	■	■				■
	<b>Модели данных. Уровень спецификации проекта</b>							
1	Диаграмма окружения атрибутов Attribute allocation diagram							
2	Диаграмма реляционных таблиц Relations diagram							
3	Домен системных атрибутов System attribute domain							
4	Системные атрибуты System attributes							

Продолжение

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP-фильтр (SAP Filter)	Фильтр для модуля ARIS for MySAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Модели данных. Уровень описания реализации</b>							
1	Диаграмма таблиц Table diagram							

## Модели процессов/управления

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP-фильтр (SAP Filter)	Фильтр для модуля ARIS for MY SAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Модели процессов / управления. Уровень определения требований</b>							
1	Карта полномочий Authorization map							
2	BSC Причинно-следственная диаграмма BSC (Balanced Scorecard) - Cause-and-effect diagram							
3	BSC Диаграмма ключевых показателей результативности BSC - Key performance indicator allocation diagram (KPI)							
4	Диаграмма управления бизнесом Business controls diagram							
5	Диаграмма классов Class diagram							
6	Диаграмма классификации Classification diagram							
7	Диаграмма взаимодействий Communications diagram							
8	Модель конкуренции Competition model							
9	Трансформация хранилища данных DW - Transformation							
10	Диаграмма сценария е-бизнеса e-Business scenario diagram							
11	Событийная цепочка процесса (eEPC) Extended event driven process chain							
12	eEPC (в виде столбцов) eEPC (column display)							
13	eEPC (уровень экземпляров) eEPC (instance)							
14	eEPC (с потоками материалов) eEPC (material flow)							
15	eEPC (в виде строк) eEPC (row display)							
16	eEPC (в виде таблицы) eEPC table display							
17	Диаграмма событий Event diagram							
18	Диаграмма окружения функции Function allocation diagram							
19	Диаграмма окружения функций (уровень экземпляров) Function allocation diagram (instance)							

## Продолжение

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP Filter - SAP-фильтр	Фильтр для модуля ARIS for MY SAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Модели процессов / управления. Уровень определения требований</b>							
20	Распределение функций по организационным уровням Function/organizational level diagram							
21	Промышленный процесс Industrial process							
22	Диаграмма потоков информации Information flow diagram							
23	Диаграмма входов/выходов Input/Output diagram							
24	Диаграмма входов/выходов (инвертированная) Input/Output diagram (invers)							
25	Карта знаний Knowledge map							
26	Диаграмма потоков материалов Material flow diagram							
27	Офисный процесс Office process							
28	OMT Модель описания классов OMT (Object Modeling Technique) Class description model							
29	OMT Декомпозиция значений данных OMT Data value decomposition							
30	OMT Динамическая модель OMT Dynamic model							
31	OMT Функциональная модель OMT Functional model							
32	OMT Объектная модель OMT Object model							
33	Диаграмма цепочки процесса (PCD) PCD (Process Chain Diagram )							
34	PCD (с потоками материалов) PCD with material flow							
35	Pipeline diagram PCD with material flow							
36	Процедурная цепочка проекта (PCD) PPC (Project process Chain)							
37	Модель инициации процесса Process instantiation model							
38	Диаграмма выбора процесса Process selection diagram							
39	Матрица выбора процесса Process selection matrix							

Продолжение

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP-фильтр (SAP Filter)	Фильтр для модуля ARIS for MySAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Модели процессов / управления. Уровень определения требований</b>							
40	Диаграмма окружения продукта Product allocation diagram							
41	Матрица выбора продукта Product selection matrix							
42	Дерево продуктов Product tree							
43	Диаграмма движения продуктов/услуг Product/Service exchange diagram							
44	Диаграмма движения продуктов/услуг (пиктографическая) Product/Service exchange diagram (graphic)							
45	Дерево продуктов/услуг Product/Service tree							
46	Дерево продуктов/услуг (пиктографическое) Product/Service tree (graphic)							
47	Диаграмма окружения роли RAD (Role assignment diagram)							
48	Модель RAMS RAMS (Requirements Analysis for Management Systems)							
49	Диаграмма прав Rights diagram							
50	Диаграмма ролей Role diagram							
51	Диаграмма правил Rule diagram							
52	SAP ALE Модель фильтров SAP ALE (application linking enabling) filter model							
53	SAP ALE Модель потоков сообщений SAP ALE message flow model							
54	SAP ALE Модель типов сообщений SAP ALE message type model							
55	Модель структуры Structuring model							
56	UML Диаграмма действий UML Activity diagram							
57	UML Диаграмма описания класса UML Class description diagram							
58	UML диаграмма класса UML Class diagram							

## Продолжение

	Модели, входящие в полный фильтр (Entire Method)	Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP-фильтр (SAP Filter)	Фильтр для модуля ARIS for MySAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
	<b>Модели процессов / управления. Уровень определения требований</b>							
59	UML Диаграмма взаимодействия UML Collaboration diagram							
60	UML Диаграмма компонент UML Component diagram							
61	UML Диаграмма состояний UML Statechart diagram							
62	UML Диаграмма использования приложений UML Use case diagram							
63	Диаграмма цепочки добавленного качества Value added chain diagram (VAD)							
	<b>Модели процессов / управления. Уровень спецификации проекта</b>							
1	Диаграмма доступа Access diagram							
2	Блок-схема программы Program flow-chart (PF)							
3	Структурная схема программы Program structure chart							
4	Модель экранного интерфейса Screen diagram							
	<b>Модели процессов / управления. Уровень описания реализации</b>							
1	Диаграмма доступа (физического) Access diagram (physical)							

**Моделирование бизнеса. Методология ARIS**

**Типы объектов**

Английское название типа	Русское название типа	Определение
Actor	Действующий субъект	Действующий субъект активизирует потоки данных, создавая либо поглощая данные, и, таким образом, является либо источником, либо потребителем в этом потоке, т.е. связан с его входом или выходом.
Application system	Прикладная система	Объект "прикладная система" конкретизирует понятие "тип прикладной системы" до уровня уникального экземпляра, идентифицируемого, к примеру, по номеру версии и/или лицензии. Компания может владеть несколькими прикладными системами (версиями) относящимися к одному типу.
Application system class	Класс прикладной системы	Схожие типы прикладных систем могут быть объединены, образуя класс прикладной системы. Принцип схожести определяется выбранным критерием классификации. Иными словами, конкретный тип прикладной системы можно отнести к нескольким классам, используя различные критерии.
Application system type	Тип прикладной системы	Объект "тип прикладной системы" отражает обобщение отдельных прикладных систем, обладающих одинаковыми техническими и функциональными характеристиками.
Association	Ассоциация	Ассоциация представляет собой группу связей, одинаковых по структуре и семантике.
Association instance	Ассоциация (экземпляр)	Ассоциация (экземпляр) отображает связи между объектами (экземплярами).
Attribute	Атрибут	Модель реляционных таблиц использует понятия "атрибут" и "реляционная таблица" для описания логической структуры базы данных. В этом случае атрибут характеризует свойства реляционной таблицы.
Attribute type group	Группа типов атрибутов	Группа типов атрибутов отображает группу ERM атрибутов одного типа сущности, тесно связанных между собой по семантике. Это позволяет, к примеру, создать группу, содержащую несколько ERM атрибутов, образующих вместе вторичный ключ.
Authorization condition	Полномочие	Данный объект можно встретить, к примеру, в моделях eEPC или PCD. Он накладывает ограничения на выполнение определенных действий организационными единицами.

## Глоссарии ARIS

Английское название типа	Русское название типа	Определение
Bitmap	Точечный рисунок	Объект, содержащий графическое изображение, которое следует поместить в соответствующую модель.
Break	Перерыв	Перерыв - объект модели "Календарь рабочих смен", обозначающий промежуток времени, в течение которого работа не выполняется. Начало и окончание перерыва должны находиться в пределах определенной рабочей смены. Используется в модуле ARIS Simulation.
Business object	Бизнес-объект	Бизнес-объект - это сложный объект, параметры которого используются для поддержки различных бизнес-процессов.
Class	Класс	Группа объектов со схожими характеристиками и отношениями образует класс. Классы отражают основные структуры моделируемого приложения.
Classification criterion	Критерий классификации	Критерий классификации определяет критерии, согласно которым следует объединить рассматриваемые объекты.
Cluster instance	Кластер (экземпляр)	Кластер (экземпляр) - это экземпляр объекта кластер/ модель данных. Он представляет собой логический взгляд на набор объектов данных или структур.
Cluster/ Data model	Кластер/ модель данных	Кластер отражает логический взгляд на набор типов сущностей и типов отношений в модели данных, что требуется для описания сложных объектов.
Column	Столбец	В модели экранного интерфейса может быть произведено графическое деление поверхности на области. Столбец соотносится с полученными таким образом вертикальными частями.
Communication	Взаимодействие	Данный объект является частью диаграммы взаимодействий и определяет тип информации, которой обмениваются две организационные единицы, а также участников этого обмена.
Company	Компания	Предприятие заданной формы собственности, действующее в рамках некоторого набора сфер деятельности.
Complex object	Сложный объект	Тип сложного объекта отображает логически связанную часть структуры модели данных.
Complex object type	Тип сложного объекта	Сложный объект - обобщение нескольких схожих сложных объектов.
Connector	Коннектор	Предназначен для отображения разделения потока данных на несколько самостоятельных потоков.
Constraint	Ограничение	Этот объект позволяет налагать ограничения на диапазоны значений для экземпляров атрибутов.

**Моделирование бизнеса. Методология ARIS**

Английское название типа	Русское название типа	Определение
Control	Управляющее воздействие	Набор мер, направленных на устранение возможных причин возникновения рисков или на снижение последствий, связанных с появлением риска.
Cost category	Категория затрат	Категория затрат отображает группу платежей одного типа, т.е. расходы, обусловленные одинаковыми причинами.
Cost driver	Стоимостной фактор	Стоимостной фактор - это единица измерения для результата (выхода) функции, например, количество обработанных заявок, потраченное на работу время и т. д.
COT attribute	COT-атрибут	Объект типа "COT- атрибут" описывает атрибуты, которые не могут быть точно определены ввиду отсутствия четкого разграничения, например "долгосрочный - среднесрочный - краткосрочный". Отдавая должное сложности содержания этих атрибутов, их следует рассматривать в качестве отдельных объектов и определять как текстовые поля ввода на экранных интерфейсах.
COT attribute (instance)	COT-атрибут (экземпляр)	COT- атрибут (экземпляр) описывает COT- атрибут в отдельно взятом случае. Иными словами, ситуация, рассмотренная на более высоком уровне в общем виде, теперь описывается более детально (пример: Клиент —> Иван Петров).
Critical factor	Критический фактор	Факторы успеха определяют те аспекты деятельности компании, которые следует принять во внимание для достижения поставленной цели. Они приписываются целям компании на диаграмме целей.
Data Store	Хранилище данных	Объект, являющийся средством хранения информации для последующего доступа к ней.
Data Value	Значения данных	Потоки данных связывают выход одного процесса (или объекта) со входом другого. Характер передаваемых данных при моделировании этих потоков описывается объектом типа "Значения данных". Объекты данного типа можно также использовать для более детального описания потоков данных в типах связей в модели "ОМТ функциональная модель" (ОМТ - техника объектного моделирования).
DBMS	СУБД	СУБД - система управления базами данных.
DBMS type	Тип СУБД	Тип СУБД отображает критерий классификации систем управления базами данных (СУБД) с одинаковыми техническими характеристиками. Как правило, типы СУБД идентифицируются по имени и номеру версии.
Documented knowledge	Документированное знание	Объект этого типа позволяет специфицировать знания, содержащиеся в документах, чертежах и относящиеся к заданной категории знаний.

## Глоссарий ARIS

Английское название типа	Русское название типа	Определение
Domain	Диапазон	С помощью объекта этого типа можно отобразить диапазон значения атрибута на модели реляционных таблиц, т.е. множество принимаемых им значений.
Domain (physical)	Диапазон (физический)	Этот объект отражает диапазон значений содержимого ячейки таблицы на диаграмме таблиц, т.е. множество принимаемых этой ячейкой значений.
Draft list	Эскиз списка	Эскиз списка определяет общую (в смысле типологии) структуру для нескольких сходных списков. Списки представляют собой интерфейсы между пользователями и прикладными системами и являются выходными данными этих систем.
Employee variable	Зарезервированная ПОЗИЦИЯ	Зарезервированное в процессе место для сотрудника, который будет специфицирован позже.
Entity	Сущность	Сущность - это реальный или абстрактный объект, представляющий интерес в рамках рассматриваемых целей и задач компании.
Entity type	Тип сущности	Типы сущности являются результатом комбинирования сходных сущностей. Сущности относятся к одному типу, если они могут быть описаны одними и теми же типовыми характеристиками (ERM-атрибутами).
Enumeration	Перечисление	Данный объект используется для конкретизации ERM-атрибутов путем перечисления их возможных типовых значений.
Enumeration list occurrence	Перечисление (экземпляр)	Является одним из значений такого объекта, как "Перечисление".
ERM attribute	ERM-атрибут	ERM-атрибуты - это характеристики, описывающие типы сущностей или типы отношений.
ERM attribute instance	ERM-атрибут (экземпляр)	ERM-атрибут (экземпляр) предназначен для описания сущности или отношения.
ERM domain	ERM-диапазон	Диапазон значений, принимаемых ERM-атрибутом.
Event	Событие	Событие - это состояние, которое является существенным для целей управления бизнесом и которое оказывает влияние или контролирует дальнейшее развитие одного или более бизнес-процессов. Изменения состояния отражаются с помощью информационных объектов. События активизируют функции и сами являются исходом выполнения функций. В отличие от функций, которые отражают процесс, протекающий во времени и имеющий определенную длительность, события происходят в одной точке во времени.

**Моделирование бизнеса. Методология ARIS**

Английское название типа	Русское название типа	Определение
Event instance	Событие (экземпляр)	Событие (экземпляр) - это конкретное событие, произошедшее при выполнении конкретного процесса. Оно идентифицируется по признаку истинности или ложности в настоящий момент, а также по абсолютному времени появления (например, событие "Подготовлен план на январь 2001 года" произошло 25 декабря 2000 года).
Field	Поле	Логическая структура СУБД может быть описана с помощью таблиц и полей. Поля определяют логическое содержание таблицы. Всевозможные значения поля могут быть описаны объектом "Диапазон (физический)".
Field (specimen)	Поле (экземпляр)	Этот объект отображает конкретный экземпляр поля таблицы. Таблицы и поля формируют логическую структуру СУБД. В компании может существовать несколько экземпляров логически определенных таблиц идентичной структуры.
Filter object type	Тип сортировки	Тип сортировки устанавливает критерии частоты (повторяемости) функций модели, выполняемых системой.
Function	Функция	Функция - некоторое действие или набор действий, выполняемых над исходным объектом (документом, материалом и т.п.) с целью получения заданного результата (документа, материала и т.п.).
Function instance	Функция (экземпляр)	Функция (экземпляр) - это конкретная функция, выполняемая в рамках конкретного процесса. С экземпляром функции связывается заданное абсолютное время начала и уникальное время окончания ее выполнения, а также другие необходимые атрибуты. В качестве примера можно привести объект "Создание плана расходов на I квартал 2001 года", который является экземпляром для функции "Создание плана расходов на квартал".
General resources	Общий ресурс	Общий ресурс - это ресурс, который не может быть отнесен к категории трудовых или операционных ресурсов. Общий ресурс является необходимым элементом для выполнения функций.
Generalization type	Тип обобщения	В процессе моделирования данных устанавливается сходство между объектами (сущностями). Эти объекты затем комбинируются, образуя тип объекта более высокого уровня обобщения.

Английское название типа	Русское название типа	Определение
Graphical user interface (GUI) type	Тип графического интерфейса пользователя	Этот объект отображает обобщение индивидуальных интерфейсов пользователя, наделенных одинаковыми характеристиками. На основе специфических типов таких интерфейсов реализуются типы прикладных систем. Тип графического интерфейса пользователя, как правило, идентифицируется по имени и номеру версии.
Group	Группа	Группа может отображать группу сотрудников, работающих вместе в течение определенного промежутка времени, например, проектная группа.
Hardware component	Компонент аппаратного обеспечения	Компонент аппаратного обеспечения - это экземпляр определенного типа оборудования, имеющегося в компании. Он часто идентифицируется при помощи инвентарного номера.
Hardware component class	Класс компонента аппаратного обеспечения	Схожие типы компонентов аппаратного обеспечения могут быть объединены, образуя класс компонента аппаратного обеспечения. Принцип схожести определяется выбранным критерием классификации. Иными словами, конкретный тип компонента аппаратного обеспечения можно отнести к нескольким классам, используя различные критерии.
Hardware component type	Тип компонента аппаратного обеспечения	Этот объект отражает обобщение отдельных экземпляров компонентов аппаратного обеспечения, обладающих одинаковыми техническими характеристиками.
Index	Индекс	Индекс позволяет сортировать поля таблицы по различным критериям, чтобы обеспечить более удобный и эффективный способ отображения ее содержимого. В диаграмме таблиц индексы могут быть связаны с полями таблицы.
Information carrier	Носитель информации	Носитель информации представляет собой средство хранения информации. Оно может быть реализовано, к примеру, в виде картотеки или компьютерных файлов.
Information flow	Поток информации	Поток информации - это объект, содержащий информацию, передаваемую, к примеру, между функцией и типом прикладной системы или между модулем и типом функции ИТ. Он используется для более точного определения связей между этими объектами и отображает данные, которыми они обмениваются.
Instantiation cycle	Цикл инициации	Этот объект принадлежит модели инициации процесса и является объектом второго (промежуточного) уровня, служащим для описания повторяющихся последовательностей интервалов инициации. Примером такого объекта является "Цикл понедельник-пятница".

**Моделирование бизнеса. Методология ARIS**

Английское название типа	Русское название типа	Определение
Instantiation interval	Интервал инициации	Этот объект принадлежит модели инициации процесса и является объектом нижнего уровня, служащим для описания самых коротких временных интервалов в потоке работ. Объект имеет следующие атрибуты: относительное время начала интервала, продолжительность интервала, количество примеров (прогонов) процесса, распределение, циклическая повторяемость, период. Пример использования этого объекта: "Для определенного процесса в 8:00 в рабочий день запустить 50 инициации (реализации) с равномерным распределением".
Instantiation plan	План инициации	Этот объект принадлежит модели инициации процесса и является объектом верхнего уровня, служащим для описания нескольких циклов инициации. Примером такого объекта является "Рабочий план утренней смены".
IT function	Функция IT	Функция IT является экземпляром типа функции. В компании может быть внедрено несколько таких экземпляров, относящихся к одному и тому же типу. Как правило, они не идентифицируются сами по себе, но их можно разделить по лицензионным номерам модулей, с которыми они связаны.
IT function class	Класс функции IT	Схожие типы функций IT могут быть объединены, образуя класс функции IT. Принцип схожести определяется выбранным критерием классификации. Иными словами, конкретный тип функции IT можно отнести к нескольким классам, используя различные критерии.
IT function type	Тип функции IT	Если рассматривать транзакции, то тип функции IT является самой маленькой единицей, относящейся к типу модулей. Он реализуется в виде отдельной части программы и выполняется целиком для завершения очередного шага обработки. Тип функции IT отражает обобщение отдельных функций IT, обладающих сходными техническими характеристиками.
Key performance indicator instance	Ключевой показатель результативности (экземпляр)	Ключевой показатель результативности является мерой измерения степени достижения цели. Он характеризуется планируемым значением, фактическим значением, а также целевым значением.
Knowledge category	Категория знаний	Этот объект используется для классификации знаний по тематикам и областям использования.
Layout	Схема оформления	Характеристики графического оформления отдельных элементов экранного интерфейса. Характеризуется цветом фона, стилем отображения текста и т. д.

## Глоссарий ARIS

Английское название типа	Русское название типа	Определение
List	Список	Списки и экраны представляют собой интерфейсы между пользователями и прикладными системами. Список - это выходные данные, предоставляемые приложением. В отличие от эскиза списка, определяющего общую (в смысле типологии) структуру для нескольких сходных списков, этот объект представляет собой конкретный экземпляр.
Location	Расположение	Расположение определяет физическое местонахождение организационных единиц, оборудования и технических ресурсов компании. Им может быть регион, город, завод, здание, комната и даже отдельное рабочее место.
Loop start	Начало цикла	Этот объект располагается на блок-схеме программы и определяет начало последовательности, состоящей из нескольких шагов. Последовательность завершается окончанием цикла. Наличие этого объекта является признаком того, что следующая за ним на схеме цепочка шагов может повторяться многократно.
Main process	Основной процесс	Представляет собой типовой процесс, свойственный нескольким сценариям на матрице выбора процессов. Вариант реализации основного процесса в рамках каждого из рассматриваемых сценариев представляется элементами тела матрицы (объекты типа "Функция" с символом "Процесс").
Material class	Класс материала	Схожие типы материалов могут быть объединены, образуя класс материалов. Принцип схожести определяется выбранным критерием классификации. Иными словами, конкретный тип материала можно отнести к нескольким классам, используя различные критерии.
Material flow	Поток материалов	Объект, отражающий движение материалов от одной функции (процесса) к другой функции (процессу).
Material type	Тип материала	Тип материала отображает обобщение материалов с одинаковыми характеристиками.
Memory location	Ячейка памяти	Ячейки памяти предназначены для хранения данных в виде таблиц или полей.
Message flow	Поток сообщений	Этот объект используется для более детального описания связей, по которым объекты обмениваются сообщениями.
Message type	Тип сообщения	Тип сообщения классифицирует потоки между системами в соответствии с передаваемыми данными и процессами, в которых они задействованы.

**Моделирование бизнеса. Методология ARIS**

Английское название типа	Русское название типа	Определение
Module	Модуль	Модуль является конкретным экземпляром типа модуля и может быть идентифицирован, например, по номеру лицензии. Компания может владеть несколькими модулями (несколькими лицензиями) одного типа.
Module class	Класс модуля	Схожие типы модулей могут быть объединены, образуя класс модуля. Принцип схожести определяется выбранным критерием классификации. Иными словами, конкретный тип модуля можно отнести к нескольким классам, используя различные критерии.
Module type	Тип модуля	Тип модуля представляет собой независимо исполняемую часть типа прикладной системы и отражает обобщение отдельных модулей, имеющих одинаковые технические характеристики.
Network	Сеть	Сеть представляет собой экземпляр, относящийся к одному из типов сетей, имеющихся в компании. Она может быть идентифицирована по своему расположению либо по инвентарному номеру.
Network class	Класс сети	Схожие типы сетей могут быть объединены, образуя класс сети. Принцип схожести определяется выбранным критерием классификации. Иными словами, конкретный тип сети можно отнести к нескольким классам, используя различные критерии.
Network connection	Сетевое соединение	Сетевое соединение является экземпляром определенного типа сетевого соединения из тех, что имеются в компании, и связывает между собой узлы сети.
Network connection type	Тип сетевого соединения	Этот объект представляет собой обобщение отдельных сетевых соединений с одинаковыми техническими характеристиками.
Network node	Узел сети	Узел сети является экземпляром определенного типа узла сети из тех, что имеются в компании. Отдельный узел представляет собой точку доступа в сеть, к которой может быть подключено какое-либо оборудование.
Network node type	Тип узла сети	Этот объект представляет собой обобщение отдельных узлов сети с одинаковыми техническими характеристиками.
Network protocol	Сетевой протокол	Сетевой протокол представляет собой стандартизованный метод связи и обмена информацией между прикладными системами в сети. Для каждого типа сети должен быть определен сетевой протокол.
Network type	Тип сети	Этот объект отражает обобщение отдельных (информационных) сетей с одинаковыми техническими характеристиками.

## Глоссарий ARIS

Английское название типа	Русское название типа	Определение
Note	Примечание	Этот объект принадлежит UML-моделям и позволяет связывать с объектами дополнительную информацию.
Object instance	Объект (экземпляр)	Конкретный экземпляр объекта, обладающий определенными параметрами.
Object type class	Класс типа объекта	Этот объект позволяет классифицировать функции, т.е. рассматривать их в соответствии с выбранным критерием.
Objective	Цель	Этот объект определяет цели организации, которые (предположительно) будут достигнуты при наличии соответствующих факторов успеха и реализации новых бизнес-процессов.
Operating resource	Операционный ресурс	Операционный ресурс является экземпляром определенного типа операционного ресурса из тех, что имеются в компании. Он часто идентифицируется по своему инвентарному номеру (например, инвентарный номер станка).
Operating resource class	Класс операционного ресурса	Схожие типы операционных ресурсов могут быть объединены, образуя класс операционного ресурса. Принцип схожести определяется выбранным критерием классификации. Иными словами, конкретный тип операционного ресурса можно отнести к нескольким классам, используя различные критерии.
Operating resource type	Тип операционного ресурса	Этот объект отражает обобщение отдельных операционных ресурсов, обладающих одинаковыми техническими характеристиками.
Operating system type	Тип операционной системы	Этот объект отражает обобщение отдельных операционных систем, обладающих одинаковыми техническими характеристиками. Операционные системы, как правило, идентифицируются по имени и номеру версии.
Operation	Операция	Операции (операторный метод) отражают функциональность или трансформации данных (атрибутов), связанные с классом.
Organizational chart	Организационная схема	Организационная схема отражает совокупность организационных взаимосвязей, рассматриваемых на верхнем уровне абстракции.
Organizational level	Организационный уровень	Организационные структуры, как правило, представляются организационными схемами. Критерием структуризации является сходство выполняемых операций, именно согласно ему формируются организационные единицы. Кроме того, структуру большой компании можно разбить на уровни, что позволяет определять ответственности и права доступа. Примерами таких уровней являются: "Область использования продукта", "Область выполнения операций" и т.п.

**Моделирование бизнеса. Методология ARIS**

Английское название типа	Русское название типа	Определение
Organizational unit	Организационная единица	Организационные единицы являются исполнителями задач, решение которых необходимо для достижения бизнес-целей. Это достаточно стабильные образования, представленные набором штатных единиц, занимаемых конкретными сотрудниками компании.
Organizational unit type	Тип организационной единицы	Данный объект отражает обобщение отдельных организационных единиц, обладающих одинаковыми характеристиками. В качестве последних могут, например, выступать права доступа и обязанности. Все подразделения компании подчиняются общим правилам, которые, благодаря наличию такого объекта достаточно определить лишь один раз.
Package	Пакет	Данный объект присутствует в моделях UML и используется для разделения элементов модели с помощью группировки компонентов и связывания их с пакетами. На событийной цепочке процесса этот объект отображает связи между входами и выходами функций как при грубой, так и при более точной детализации, т.е. отображает только источник или использование информации. Пакет не обладает функциями чтения или записи.
Packaging material class	Класс упаковочного материала	Типы материалов могут быть объединены в классы упаковочных материалов. При таком объединении используются различные критерии классификации. Другими словами, один тип упаковочного материала может относиться к нескольким классам материалов.
Packaging material type	Тип упаковочного материала	Тип упаковочного материала - это совокупность отдельных упаковочных материалов, имеющих одинаковые характеристики.
Page	Страница	Данный объект делит экранный интерфейс на страницы.
Parameter	Параметр	Данный объект отображает входные и выходные параметры функции ИТ.
Person	Сотрудник	Сотрудник является отдельным служащим компании (идентифицируемым, к примеру, по его персональному коду) и может быть связан с организационными единицами (в которые он входит), а также с функциями (которые он исполняет или за которые отвечает).

## Глоссарии ARIS

Английское название типа	Русское название типа	Определение
Person type	Тип сотрудника	Данный объект отображает обобщение отдельных сотрудников, имеющих одинаковые характеристики. Такими характеристиками могут быть права доступа и обязанности. Например, ответственность начальников отделов, в случае существования такого объекта, потребуется описать только один раз.
Perspective	Точка зрения	Точка зрения, с которой рассматривается организация в процессе стратегического планирования.
Position	Должность	Элементарной организационной единицей компании является должность. С ней связаны сотрудники, и, как правило, их права и обязанности определяются именно профилем должности.
Process	Процесс	Процесс отображает функцию, преобразующую значения данных. Результат процесса зависит от поведения всей системы в целом.
Product/Service	Продукт/услуга	Выходом процесса-цепочки добавленного качества является продукт/услуга. Он представляет собой результат человеческих действий или технического процесса и может являться как материальным продуктом (тип материала, тип операционного ресурса, вспомогательные технические средства, тип упаковочного материала), так и услугой. Причиной для производства продукта/услуги является требование организационной единицы компании или клиента.
Program library	Библиотека приложений	Библиотека приложений содержит программы и/или программные модули, требуемые для выполнения задачи.
Program module	Программный модуль	Программный модуль - это отдельный программный файл, расположенный на носителе данных (например, файл EXE или COM) и приобретаемый вместе с лицензией.
Program module type	Тип программного модуля	Тип программного модуля отражает обобщение отдельных программных модулей, обладающих одинаковыми техническими характеристиками.
Programming language	Язык программирования	Язык программирования представляет собой набор команд и синтаксических правил для создания программ. Чаще всего, он может быть идентифицирован по номеру лицензии.
Relation	Реляционная таблица	Реляционная таблица описывает тип сущности через ее атрибуты. Это подмножество всевозможных комбинаций диапазонов значений атрибутов.
Relationship	Отношение	Отношение - это логическая связь между сущностями. Схожие отношения могут быть объединены, образуя тип отношения.

**Моделирование бизнеса. Методология ARIS**

Английское название типа	Русское название типа	Определение
Relationship type	Тип отношения	Отношение - это логическая связь между сущностями. Схожие отношения могут быть объединены, образуя тип отношения. Отношения относятся к одному типу, если их можно описать с помощью одинаковых атрибутов.
Risk	Риск	Риск означает потенциальную опасность для процесса не достигнуть желаемой цели
Rule	Правило	Данный объект представляет собой правило разветвления и слияния веток процесса. Если перейти к рассмотрению каждой отдельной функции процесса, то можно сказать, что правило отражает логическое соотношение между несколькими исходными для функции событиями и несколькими результирующими.
Rule instance	Правило (экземпляр)	Данный объект представляет собой логический оператор уровня экземпляров и определяет связь между функцией и событием на этом уровне.
Screen	Экранный интерфейс	Объект представляет собой интерфейс между пользователем и прикладной системой, визуализируя процесс ввода и получения данных. В отличие от типа экранного интерфейса, описывающего несколько схожих интерфейсов, данный объект является конкретным экземпляром такого типа.
Screen design	Таблица экранного интерфейса	Перечисляет объекты и типы объектов на экранном интерфейсе.
Section	Раздел	Видимая область на модели экранного интерфейса может быть разделена на несколько частей. Данный объект описывает полученные таким образом ряды.
Separator	Разделитель	Пустой ряд или столбец на экранном интерфейсе.
Shift	Смена	Объект модели "Календарь смен", определяющий интервал времени, в течении которого материальные и трудовые ресурсы доступны для выполнения функций. С его помощью можно отражать дневные и ночные смены и т.д. Если смена не связана с каким либо циклом рабочих смен, то она повторяется каждые 24 часа с перерывами, определенными объектом "Перерыв". Используется в модуле ARIS Simulation.

## Глоссарий ARIS

Английское название типа	Русское название типа	Определение
Shift cycle	Цикл смен	Объект модели "Календарь смен", определяющий период работы каждой рабочей смены и конкретные рабочие дни. Существует возможность задать одно- или двухнедельный цикл. Используется в модуле ARIS Simulation.
Shift plan	План смен	Объект модели "Календарь смен", объединяющий циклы рабочих смен. Показывает, когда и какие циклы повторяются. Служит основой при определении последовательности утренних, вечерних и особых рабочих смен. Используется в модуле ARIS Simulation.
Solution	Решение	Решение - это продуманный комплекс управляющих воздействий, обеспечивающий полную нейтрализацию одного или нескольких рисков.
Sp./Gen. operator (specialization/generalization operator)	Оператор обобщения/специализации	Оператор обобщения/специализации разделяет или комбинирует схожие классы.
State	Состояние	Некоторый набор значений, описывающих реакцию объекта на совокупность событий (воздействий).
Strategy	Стратегия	Стратегия представляет собой самодостаточную совокупность стратегических целей, направленную на достижение компанией заданного состояния.
Structural element	Структурный элемент	Этот объект используется для классификации моделей в соответствии с различными точками зрения, например, по функциональному признаку, с точки зрения процесса или при сертификации по стандарту ИСО 9001.
System attribute	Системный атрибут	Системные атрибуты описывают объекты, связанные с задачей экспорта данных. Возможно отображение первичных и внешних ключей, обязательных и описательных полей.
System attribute domain	Диапазон системного атрибута	Диапазон системных атрибутов описывает область допустимых значений для системных атрибутов.
System organization unit	Системная организационная единица	Интегрированные прикладные системы содержат организационные структуры, которые следует принять во внимание при внедрении таких систем. Данный объект используется при моделировании этих организационных структур.

**Моделирование бизнеса. Методология ARIS**

Английское название типа	Русское название типа	Определение
System organization unit type	Тип системной организационной единицы	Этот объект представляет собой обобщение отдельных системных организационных единиц с похожими характеристиками. Схожесть может, например, заключаться в правах доступа к прикладной системе.
Table	Таблица	Логическая структура СУБД может быть описана с помощью таблиц и полей. Таблица представляет собой обобщение отдельных экземпляров таблиц, имеющих одинаковую логическую структуру.
Tables (specimen)	Таблица (экземпляр)	Данный объект представляет собой отдельный экземпляр таблицы, которая, наряду с полями, описывает логическую структуру СУБД. В организации может существовать несколько экземпляров логически определяемых таблиц и полей, имеющих одинаковую структуру.
Technical operating supply	Вспомогательные технические средства	Вспомогательные технические средства являются экземпляром определенного типа вспомогательных технических средств.
Technical operating supply class	Класс вспомогательных технических средств	Класс вспомогательных технических средств - это объединение на основе определенного критерия классификации схожих типов вспомогательных технических средств.
Technical operating supply type	Тип вспомогательных технических средств	Этот объект отражает обобщение отдельных экземпляров вспомогательных технических средств, обладающих одинаковыми техническими характеристиками.
Technical term	Технический термин	Технический термин отражает концептуальный взгляд на имеющиеся в организации информационные объекты и используется для выделения специфичных терминов и понятий, а также существующих между ними взаимосвязей.
Technical term instance	Технический термин (экземпляр)	Этот объект представляет собой технический термин на уровне экземпляров.
Text	Текст	Объект этого типа содержит общее текстовое описание, включенное в соответствующую модель.
Transaction folder	Папка транзакций	Папка транзакций представляет собой набор транзакций, передаваемых для обработки.

## Глоссарий ARIS

Английское название типа	Русское название типа	Определение
Transport systems	Транспортная система	Транспортная система предназначена для перевозки грузов и используется в качестве ресурса для таких функций компании, как доставка продукции на склады клиентов, осуществление самостоятельного вывоза материалов и оборудования и так далее. Она является экземпляром определенного типа транспортной системы и часто идентифицируется по своему инвентарному либо заводскому номеру.
Transport system class	Класс транспортной системы	Класс транспортной системы - объединение схожих типов транспортных систем. Объединение происходит на основе выбранного критерия классификации.
Transport system type	Тип транспортной системы	Этот объект отражает обобщение отдельных экземпляров транспортных систем, обладающих одинаковыми техническими характеристиками.
Use case	Пример использования	Этот объект присутствует в моделях UML и описывает случаи использования, т.е. те случаи, при которых действующий субъект совершает определенные действия.
View	Взгляд	Объект отражает логический взгляд на набор реляционных таблиц и атрибутов. К примеру, представление менеджера по продажам о понятии "Заказ" может отличаться от видения кладовщика.
View (physical)	Взгляд (физический)	Объект отражает логический взгляд на набор таблиц и полей.
Warehouse equipment	Складское оборудование	Складское оборудование является экземпляром определенного типа складского оборудования из тех, что имеются в компании и представляет собой средства погрузки и хранения продукции на складе. Оно часто идентифицируется по своему номеру.
Warehouse equipment class	Класс складского оборудования	Этот объект отражает обобщение отдельных экземпляров складского оборудования, обладающих одинаковыми техническими характеристиками.
Warehouse equipment type	Тип складского оборудования	Класс складского оборудования - объединение схожих типов складского оборудования. Объединение происходит на основе выбранного критерия классификации.

## Типы связей

Английское название	Русское название
accepts	утверждает результат
accesses alternatively	имеет альтернативный доступ к
accesses	имеет доступ к
activates (Baan)	активизирует(Baan)
activates	активизирует
affects	влияет
aggregates	агрегирует
allocation of cost centers	назначение центров затрат
applies to	использует
approves	утверждает
archives	помещает в архив
assigns	связывает
associates (multiple)	ассоциируется с (множественная связь)
associates	ассоциируется с
belongs to (Baan)	принадлежит (Baan)
belongs to class	принадлежит классу
belongs to type	принадлежит типу
belongs to	принадлежит
calls	вызывает
can access	может иметь доступ к
can be accessed by	может быть доступен для
can be accessed	может быть доступен для
can be assigned to	может быть связан(а) с
can be constituent	может являться частью
can be created by	может создаваться

## Глоссарий ARIS

Английское название	Русское название
can be disciplinary superior	может являться непосредственным руководителем
can be end point of	может являться конечной точкой для
can be instantiated by	может обрабатываться
can be located at	может быть расположен в
can be location of	может являться местом расположения для
can be platform for	может являться аппаратной платформой для
can be processed by/with	может выполняться с использованием
can be reached by	может быть получен с использованием
can be realized at	может быть реализован в
can be responsible for	может являться ответственным за
can be supported by	может поддерживаться
can be technical superior	может являться техническим руководителем
can be under responsibility of	может находиться под ответственностью
can be used by	может использоваться
can be user	может являться пользователем
can be	может являться
can belong to	может принадлежать
can change workflow to	может направлять поток работ к
can consist (hor.) of	может состоять из (горизонт.)
can consist (vert.) of	может состоять из (верт.)
can create	может создавать
can end in	может заканчиваться в
can have as constituent	может содержать
can have as employee	может иметь в качестве сотрудника
can have disciplinary superior	может находиться под непосредственным руководством
can have technical superior	может находиться под техническим руководством
can occur	может относиться к

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
can realize	может являться реализацией
can replace	может заменять
can run on	может быть запущен на
can run under	может быть запущен на
can subsume part of connection	может являться частью соединения
can subsume section	может являться секцией
can subsume	может являться обобщающей категорией для
can support	может поддерживать
can use	может использовать, может представляться
changes	изменяет
checks	проверяет
classified by	классифицируется
classifies	классифицирует
communicates with	взаимодействует с
composes	формирует
concerns	имеет отношение к
consists (hor.) of	состоит из (гор.)
consists (vert.) of section	состоит из секций (верт.)
consists of	состоит из
constrains	накладывает ограничения на
consults	консультирует
consumes	потребляет
contains (Baan)	содержит (Baan)
contains	содержит
contributes to	способствует при выполнении
cooperates with	взаимодействует с
corresponds to	соответствует

## Глоссарий ARIS

Английское название	Русское название
cost category allocation	окружение стоимостной категории
creates output to	создает на выходе
creates	порождает
decides on	принимает решение по
defines (hierarchical)	определяет(иерархия)
defines an association	определяет ассоциацию
defines screen field	определяет поле
defines status	определяет статус
deletes	уничтожает
denotes maturity level of	указывает на степень завершенности
depends	зависит от
depicts	описывает
designs	проектирует
determines type of	является типом для
determines volume of	определяет объем
development is under responsibility of	разрабатывается под руководством
differentiates according to value of	дифференцируется в зависимости от
distributes	распределяет
divides	делит
documents	документирует
does not consume	не потребляет
ends in	оканчивается в
error (in)	ошибка (вход)
error (out)	ошибка (выход)
evaluates function attribute	обрабатывает атрибут функции
evaluates	оценивает
executes	выполняет

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
extend	расширяет
follows	следует за
forms a reference with	является ссылочным для
generalizes	обобщает
gets input from	получает входные данные из
groups	группирует
has access by	доступна для
has alternative operating resource	имеет в качестве альтернативного ресурса
has as employee	имеет в качестве служащего
has as subtype	имеет подтип
has as supertype	имеет супертип
has assigned	связан(а) с
has associated	связывается с
has attribute type group	имеет группу типов атрибутов
has attribute	имеет атрибут
has condition (false)	имеет условие (ложное)
has condition (true)	имеет условие (истинное)
has consulting role in	участвует в качестве консультанта
has critical factor	имеет критический фактор
has current location	расположен в (текущее расположение)
has domain	имеет область
has employee variable	имеет временного служащего
has executed	выполнил
has external key	имеет внешний ключ
has feature	имеет свойство
has full influence on	имеет полное влияние на
has influence on	имеет влияние на

## Глоссарий ARIS

Английское название	Русское название
has information about	отображает содержание
has input of	имеет на входе
has input parameter	имеет входной параметр
has instance	имеет экземпляр
has interaction from	взаимодействует с (пассив)
has knowledge of	владеет знанием
has link to	имеет связь с
has mandatory field	имеет обязательное поле
has member	имеет в своем составе
has method	имеет метод
has minimal influence on	оказывает минимальное влияние на
has normal influence on	оказывает обычное влияние на
has operating resource	имеет в качестве ресурса
has operation	имеет операцию
has order basis	имеет в качестве основы для упорядочения
has output of	имеет на выходе
has output parameter	имеет выходной параметр
has owner	имеет в качестве владельца
has part	состоит из
has performance characteristic	имеет в качестве характеристики выполнения
has performance component	имеет в качестве исполняемого компонента
has position	имеет должность
has possible detour to	имеет возможный запасной путь
has primary key	имеет первичный ключ
has process manager	имеет менеджера процесса
has properties supplied by	имеет свойства, полученные от
has quota of	имеет квоту

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
has recursion	имеет рекурсию
has reference to	ссылается на, имеет ссылку на
has relation with	имеет отношение к
has relationship to	имеет отношение к
has root	имеет корень
has screen field	имеет поле
has session	имеет сессию
has sink	имеет приемник
has source	имеет источник
has specimen owner	имеет в качестве владельца экземпляра
has specimen	имеет в качестве экземпляра
has state	отображает
has strong influence on	сильно влияет на
has subordinate event	имеет зависимое событие
has subordinate rule	имеет зависимое правило
has subprocess	имеет подпроцесс
has substate	имеет промежуточное состояние
has subtype	имеет подтип
has supertype	имеет супертип
has the disciplinary superior	имеет в качестве непосредственного руководителя
has the technical superior	имеет в качестве технического руководителя
has transition from	состояние после перехода из
has transition to	переходит в
has user	используется (кем)
has value	имеет значение
has	имеет
helps with the creation of	создается с помощью

## Глоссарий ARIS

Английское название	Русское название
implements	реализует
imports	импортирует
include	включает
influences	влияет на
Information flow (passive)	информационный поток (пассив)
Information flow	информационный поток
instantiate	иллюстрирует
interacts with	взаимодействует с
interrupts	прерывает
is a base for	является базой для
is a component of	является составной частью
is a functional generic term of	является практическим обобщающим термином для
is a functional subterm of	является практическим подтермином для
is a receiver of	получается от
is a	является
is accepted by	результат утверждается (кем)
is accessed alternatively by	доступен(ы) (альтернативный вариант) для
is accessed by	доступен для
is accessed	доступен для
is activated by (Baan)	активизируется (Baan)
is activated by	активизируется
is affected by shift	зависит от смены
is affected by	подвержено влиянию
is aggregating	является подмножеством
is alternative operating resource of	является альтернативным ресурсом для
is approved during	утверждается в ходе выполнения
is archived by	архивируется

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
is assigned 1:1	имеет отношение 1:1
is assigned 1:n	имеет отношение 1:n
is assigned m:n	имеет отношение m:n
is assigned n:1	имеет отношение n:1
is assigned n:m	имеет отношение n:m
is assigned to type	относится к типу
is assigned to	связан(а) с
is associated by (multiple)	имеет в качестве ассоциации (множественная связь)
is associated with	ассоциируется с
is associated	имеет в качестве ассоциации
is attribute of	является атрибутом для
is attribute type group	является группой типов атрибутов для
is available at	может быть получено от/из
is based on	основывается на
is by default processed by	по умолчанию обрабатывается с помощью
is called by	вызывается
is changed by	изменяется
is characteristic for	является характеристикой для
is checked by	проверяется
is classified	классифицируется
is compared to	сравнивается с
is composed of	состоит из
is concerned by	затрагивается
is constrained	ограничена
is consumed by	потребляется
is consumed	потребляется
is contained by	содержится в

## Глоссарий ARIS

Английское название	Русское название
is contained in	содержится в
is contents of	содержится в
is created by	порождается
is created with help from	помогает в создании
is criterion of	является критерием для
is critical factor for	является критическим фактором для
is current location of	является текущим местом расположения
is decided by	зависит от решения
is defined by	определяется
is defined	определяется
is deleted by	уничтожается
is dependent on	порождается через
is depending hierarchically	подчинен по иерархии
is depending	влияет на
is depicted by	описывается
is described by	описывается
is describing for	описывает
is designed by	проектируется
is developed with	пишется на
is different from	отличается от
is disciplinary superior to	является непосредственным руководителем
is discriminated by	дифференцируются по
is discriminator of	дифференцирует
is distributed by	распределяется
is divided into	делится на
is documented in	документировано в
is domain for	является диапазоном для

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
is driver	является драйвером для
is effect of	является результатом
is employee variable	является временным служащим
is end of	является концом для
is evaluated by	оценивается с помощью
is executed at	выполняется на территории
is executed by	выполняется
is execution-oriented subordinate	подчиняется по способу выполнения
is execution-oriented superior	подчиняет по способу выполнения
is exposed to	раскрывается на
is extended	расширяется
is external key for	является внешним ключом для
is feature of	является свойством
is filter of	является фильтром для
is filtered by	фильтруется
is financially responsible for	является ответственным за финансовые вопросы
is fully influenced by	полностью зависит от
is generalization of	является обобщением
is generic term of	является обобщающим термином для
is implemented by	реализуется в виде
is imported	импортируется
is in conflict with	находится в противоречии с
is included	включается в
is influenced by	зависит от
is input for	является входом для
is input parameter of	является входным параметром для
is instance of	является экземпляром

## Глоссарий ARIS

Английское название	Русское название
is instance	является экземпляром
is instantiated by	запускается при помощи
is interrupted by	прерывается
is IT responsible for	отвечает по IT за
is linked by	следует за
is linked to	связан с
is linked with	связан с
is located at	располагается
is managed by	находится под управлением
is managed with	управляется с помощью
is mandatory field for	является обязательным полем для
is mandatory field	является обязательным полем
is measured by	измеряется с помощью
is member of	является составной частью
is method for	является методом для
is minimally influenced by	в минимальной степени зависит от
is nested	является вложением
is normally influenced by	в нормальной степени зависит от
is not consumed	не потребляется
is object-oriented subordinate	подчиняется по объекту
is object-oriented superior	подчиняет по объекту
is occupied by	занята
is of type	относится к типу
is operating resource of	является ресурсом для
is operation of	является операцией для
is operationalized	обрабатывается
is order basis for	является основой для упорядочения

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
is organization manager for	является организационным управляющим
is oriented at	ориентируется по
is output medium for	является выходной информацией
is output of	является выходом для
is output parameter of	является выходным параметром для
is owner of	является владельцем для
is parallel to	является аналогом для
is part of	является частью
is part of	является частью
is partly consumed	частично потребляется
is performance characteristic of	является характеристикой производительности для
is performance component of	является компонентом производительности для
is performed by	формируется
is performed in	является местом выполнения
is platform of	является платформой для
is position of	является должностью
is predecessor of	предшествует
is prevented by	предотвращается
is primary key for	является первичным ключом для
is process manager for	является менеджером процесса для
is process manager	является менеджером процесса
is process-oriented subordinate	подчиняется по процессу
is process-oriented superior	подчиняет по процессу
is produced by	производится
is programming language of	является языком программирования для
is propagated by	передаются
is propagated	передается

## Глоссарий ARIS

Английское название	Русское название
is provided by	обеспечивается
is reached via	достигается посредством
is read by	считывается
is realized by	реализуется
is received from	направляется к
is reduced by	сокращается посредством
is referenced by	является ссылкой для
is repeated by	повторяется
is reported by	документирует
is represented by	представляется
is represented in	отображается на
is required alternatively	требуется(альтернативный вариант)
is required for	требуется
is responsible for development of	отвечает за разработку
is responsible for	является ответственным за
is sent from	направляется от
is set by	устанавливается (кем)
is shifted to	переключается на
is source of	является источником
is specialization of	является частным случаем
is specimen of	является экземпляром
is specimen owner of	является владельцем экземпляра
is splitted by	разделяются
is state of	отображается в
is stored on	хранится
is strongly influenced by	сильно зависит от
is subordinate event of	является зависимым от

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
is subordinate rule of	является зависимым от
is subordinate	подчиняется
is subprocess of	является подпроцессом
is substate	является промежуточным состоянием
is substituted by	замещается
is subterm of	является подтермином для
is subtype of	является подтипом
is successor of	следует за
is Superior (Baan)	имеет в подчинении (Baan)
is superior	имеет в подчинении
is supertype of	является супертипом
is supported by consulting role of	выполняется при консультировании
is supported by using	поддерживается при использовании
is supported by	поддерживается
is supported	поддерживается
is taken at	используется
is technical superior to	является техническим руководителем
is technically responsible for	отвечает за техническую часть
is under financial responsibility of	отвечает по финансам за
is under IT responsibility of	имеет в качестве ответственного по ИТ
is under organizational responsibility of	находится под организационным руководством
is under process responsibility of	находится под управлением
is under responsibility of	находится под ответственностью
is under technical responsibility of	имеет в качестве ответственного за техническую часть
is used by	используется
is used in	используется в
is used	используется

## Глоссарии ARIS

Английское название	Русское название
is user interface	является интерфейсом пользователя для
is user	является пользователем
is value of (Baan)	является значением (Baan)
is value of	является значением
is worked on by collaboration of	выполняется при сотрудничестве с
is workflow class of (passive)	является классом workflow (пассив)
is workflow class of	является классом workflow
is	является
leads to (Baan)	формирует(Baan)
leads to	порождает событие через
lies on	отображается
links	предшествует
manages	управляет
Material flow (passive)	поток материалов (пассив)
Material flow	поток материалов
may access	может иметь доступ к
may be executed by	может быть выполнена
may change privileges (passive)	права доступа могут быть изменены (кем)
may change privileges	может изменять права доступа к
may execute	может выполнять
may instantiate	может подвергать обработке
may not be executed by	может быть не выполнена
may not execute	может не выполнять
may search (passive)	может использоваться при поиске
may search	может осуществлять поиск
measures	измеряет
mitigates	сокращает

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
must inform about result of	должен информировать о результатах выполнения
must not occur for	не должно произойти
must occur for	должно произойти
nests	содержит
next	следующий
no (in)	нет (вход)
no (out)	нет (выход)
occupies	занимает
occurs after	происходит после
occurs before	происходит до
Optimization Path (passive)	путь оптимизации (пассив)
Optimization Path	путь оптимизации
owns	содержит
partly consumes	частично потребляет
passes value	передает значение
performs	формирует
Phase assignment (Baan)	детализация фазы (Baan)
Phase assignment	детализация фазы
prevents	предотвращает
previous	предыдущий
processes by default	обрабатывает по умолчанию
produces	производит
propagates	передает
provides input for	предоставляет входные данные для
provides value for	предоставляет значение для
provides	обеспечивает
queries	запрашивает

## Глоссарии ARIS

Английское название	Русское название
reads	считывает
realizes	реализует
receives	получает от
references	ссылается на
relates to (passive)	имеет отношение к (пассив)
relates to	имеет отношение к
repeats	повторяет
represents	отображает
requires alternatively	требует (альтернативный вариант)
requires	требует
result is forwarded by	результаты выполнения передаются (кем)
Role Assignment (passive)	детализация роли (пассив)
Role Assignment	детализация роли
runs under	запускается на
sees	считывает
sends information on cancellation to	информация о нестандартном завершении передается (кому)
sends	направляет в
sets value	устанавливает значение
specializes	специализирует
splits to	разделяет
stores	хранит
substitutes for	замещает
subsumes part of connection	является частью соединения
subsumes section	является секцией
subsumes	содержит, объединяет

**Моделирование бизнеса. Методология ARIS**

Английское название	Русское название
supplies properties for	предоставляет свойства для
supports when time limit is exceeded (passive)	поддерживается при превышении ограничения времени (пассив)
supports when time limit is exceeded	поддерживает при превышении ограничения времени
supports	поддерживает
synonyme (not a preferred term of)	синоним (нежелательный термин)
synonyme (preferred term of)	синоним (предпочтительный термин)
takes	берет
time assessment position	состояние временной оценки
uses value of	использует значение
uses	использует
volume is dependent on	объем зависит от
was executed by	было выполнено (кем)
yes (in)	да (вход)
yes (out)	да (выход)

## Список литературы

1. *Шеер А.-В.* Бизнес-процессы. Основные понятия. Теория. Методы. — М.: Весть-МетаТехнология, 1999.
2. *Шеер А.-В.* Моделирование бизнес-процессов. — М.: Весть-МетаТехнология, 2000.
3. *Hammer M., Champy J.* Reengineering the corporation: A manifesto for business revolution. - N.Y.: Harper-Collins, 1993.
4. Автоматизированные информационные технологии в экономике / Под ред. Г.А. Титоренко. — М.: Компьютер, ЮНИТИ, 1998.
5. *Спицнадель В.Н.* Основы системного анализа. — СПб.: Изд. дом «Бизнес-пресса», 2000.
6. *ХанД.* Планирование и контроль: концепция контроллинга. — М., Финансы и статистика, 1997.
7. *Марка Д., МакГоуэн К.* Методология структурного анализа и проектирования. — М.: МетаТехнология, 1993.
8. *Калянов Г. Н.* Консалтинг при автоматизации предприятий: Научно-практическое издание. Серия «Информатизация России на пороге XXI века». — М.: СИН-ТЕГ, 1997.
9. *Лебедев А.Н.* Моделирование в научно-технических исследованиях. — М.: Радио и связь, 1989.
10. *Ойхман Е.Г., Попов Э.В.* Реинжиниринг бизнеса: Реинжиниринг организаций и информационные технологии. — М.: Финансы и статистика, 1997.
11. ARIS Methods. Version 5. June 2000. Copyright (©) 1997 - 2000 by IDS Scheer AG, Saarbrucken.
12. ARIS Quick Start Guide. Version 5. May 2000. Copyright (©) 1997 - 2000 by IDS Scheer AG, Saarbrucken.
13. POET© Object Server for ARIS 5 May 2000. Copyright (©) 1997 - 2000 by IDS Scheer AG, Saarbrucken.
14. ARIS Installation Guide. Version 5. April 2000. Copyright (©) 1997 - 2000 by IDS Scheer AG, Saarbrucken.
15. *Chen P.P.* The Entity-Relationship Model: Toward a Unified View of Data. — ACM Transactions on Database Systems, vol. 1, № 1, 1976.
16. Терминология и глоссарий WfMC. Спецификация коалиции по управлению workflow (Workflow management coalition). — М.: Весть-МетаТехнология, 2000.
17. *Кулопулос Т. М.* Необходимость workflow. Решения для реального бизнеса. — М.: Весть-МетаТехнология, 2000.
18. *Фишер Л.* Совершенство на практике. Лучшие проекты в области управления бизнес-процессами и workflow. — М.: Весть-МетаТехнология, 2000.
19. ИСО 9000-2000. Системы менеджмента качества. Основные положения и словарь.
20. ИСО 9001-2000. Системы менеджмента качества. Требования.
21. ИСО 9004-2000. Системы менеджмента качества. Рекомендации по улучшению деятельности.
22. ИСО 10006-1997. Менеджмент качества. Руководство качеством при управлении проектами.