

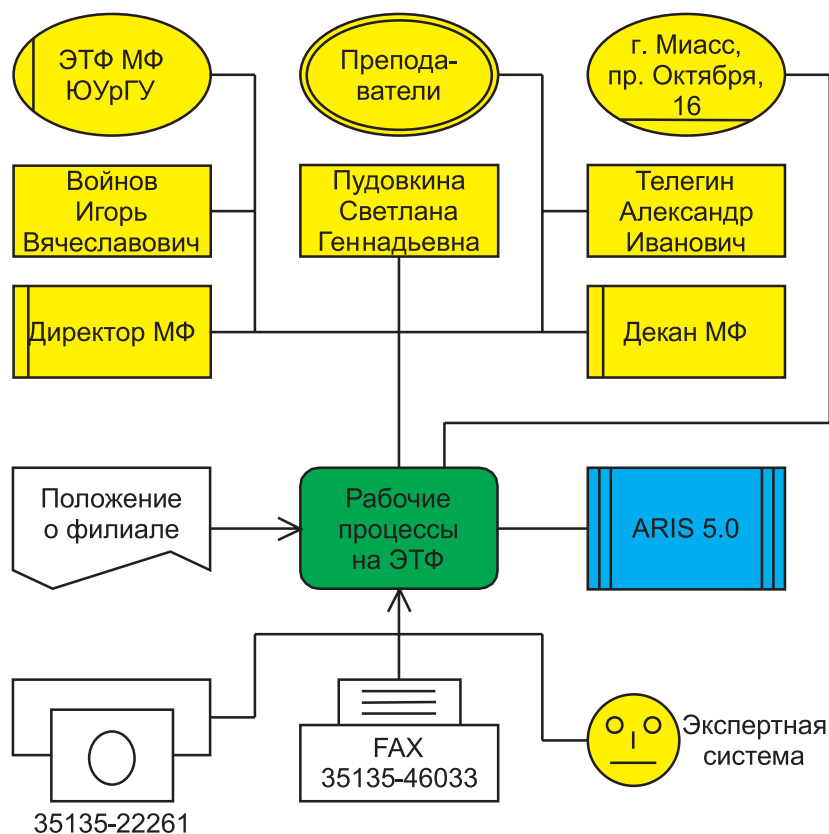
Уральское отделение РАН  
Южно-Уральский государственный университет  
Миасский научно-учебный центр

И. В. Войнов, С. Г. Пудовкина, А. И. Телегин

# ***МОДЕЛИРОВАНИЕ ЭКОНОМИЧЕСКИХ СИСТЕМ И ПРОЦЕССОВ***

***ОПЫТ ПОСТРОЕНИЯ ARIS-МОДЕЛЕЙ***

**МОНОГРАФИЯ**



Челябинск  
2002

УДК 519.86+681.518  
ББК У.в6

Войнов И. В., Пудовкина С. Г., Телегин А. И. Моделирование экономических систем и процессов. Опыт построения ARIS-моделей: Монография. – Челябинск: Изд. ЮУрГУ, 2002. – 392 с.

В книге рассмотрены методы моделирования сложных систем и процессов, в них происходящих. В качестве примеров использования изложенных методов построены более 100 статических и динамических моделей экономических систем и процессов. Описана роль и функции информационных технологий в экономике. Изложены области практического применения бизнес-моделей. Построена математическая модель финансово-обеспеченного производства фирмы. Приведены примеры эффективного применения математических методов в экономике и измерения бизнес-процессов. Рассмотрены основы построения ARIS-моделей и практически в полном объеме изложена методика построения всех типов ARIS-диаграмм в программной среде ARIS 5.0 с простейшим методологическим фильтром.

Книга предназначена специалистам в области моделирования сложных систем и процессов. Особенно полезной она будет тем, кто применяет CASE-средства для моделирования, анализа, оптимизации и автоматизации бизнес-процессов и, в частности, использует для этого ARIS-методологию и одноименную программную систему. Книгу можно использовать в качестве учебного пособия по ряду дисциплин специальности “Прикладная информатика в экономике”. Она будет полезна менеджерам, преподавателям, аспирантам и студентам экономических специальностей как учебное пособие по теории бизнес-информатики.

Ил. 224, табл. 46, список лит. – 29 назв.

Редакционная коллегия:

Баев И.А., Верзакова Г.А., Вяткин Г.П., Глазунов П.С., Григориади В.С., Гуревич С.Ю., Дегтярь В.Г., Дерябин В.Д., Ершов Н.П., (главный редактор), Ершов П.Н. (ответственный секретарь), Зиновьев Р.С., Максимов Ю.В., Панов В.В., Радашкевич М.М., Соколовский М.И., Стружанов В.В., Уфимцев С.А.

Рецензенты: д.т.н., проф. Тихонов Н. Н.,  
к.э.н., доцент Криничанский К. В.

Ответственный за выпуск Верзакова Г.А.

Литературный редактор Коваль М.Л.

Компьютерная верстка и дизайн Максимов В. И.

Печатается по решению Миасского научно-учебного центра

ISBN 5-696-01832-7

© ЮУрГУ, 2002.



# ***ВВЕДЕНИЕ***

*Для раскрытия назначения этой книги ответим на следующие вопросы:*

*Зачем нужны модели и моделирование?*

*Почему модели имеют графический (диаграммный) вид?*

*Где применяются CASE-средства?*

*Для чего используются модели?*

*Какие особенности у функционального и объектно-ориентированного моделирования?*

*Почему выбирают АРИС-моделирование?*

*Чем хороша программная среда ARIS 5.0 (e-Business Suite)?*

*Что изложено в этой книге?*

*Кому рекомендуется эта книга?*

*Что мы использовали и кто нам помог?*

*Перечисленные вопросы возможно задаст себе наш потенциальный читатель, прежде чем решиться купить эту книгу и потратить время на ее изучение. Надеемся, что изложенные ответы помогут ему сделать правильный выбор.*

## ***Зачем нужны модели и моделирование?***

Теория моделирования технических систем и процессов в 20-м веке убедительно доказала свою эффективность и практическую значимость. Этого нельзя сказать о моделировании экономических систем и процессов. Достижения в области математики и быстрое развитие средств вычислительной техники существенно расширили возможности создания и развития новых высокоэффективных теорий моделирования. Методы расчета и конструирования сложных технических систем, основанные на моделировании, а также эксплуатация систем, реализованных на основе этих расчетов, подтверждают совпадение моделей и реально созданных изделий. Аналогичных примеров практического использования моделей экономических систем в мировой практике немного, а в российской действительности они пока единичны. Но заказчик, например, автоматизированной системы управления предприятием или системы автоматизации технологических процессов желает “увидеть” и скорректировать будущую систему до того, как он оплатит ее изготовление и она будет реализована физически. А для этого необходимо моделирование проектируемой системы.

Термин “моделирование” применяется в двух смыслах. Во-первых, под моделированием понимают процесс построения модели системы. Во-вторых, под моделированием понимают процесс исследования модели функционирования системы. Таким образом, если модели еще нет, то моделирование – это процесс построения модели, если модель уже есть, то моделирование – это процесс имитации функционирования

системы на этой модели. Когда говорят “имитационное моделирование”, то однозначно понимают второй смысл термина “моделирование”. Процесс имитации, как правило, осуществляют с целью исследования поведения системы, ее анализа и, в конечном счете, для разработки планов улучшения этой системы.

Наличие комплексной модели предприятия является основой для выполнения следующих работ: проведения анализа, оценки и внесения предложений по совершенствованию деятельности предприятия; разработки автоматизированной системы управления предприятием; разработки системного проекта и внедрения корпоративной информационной системы (КИС), поддерживающей систему управления; подготовки и проведения процедуры сертификации предприятия в соответствии с требованиями международных стандартов качества серии ИСО 9000 и т. д.

## ***Почему модели имеют графический (диаграммный) вид?***

К простейшим моделям в экономике относятся производственные функции, которые описывают зависимость результатов производства от одного или нескольких ресурсов (природных, трудовых, финансовых, производственных, информационных). Известно, что функциональную зависимость можно описать словесно, аналитически, таблично или графически. Но для словесного задания функции нужно быть хорошим “писателем”, а аналитический вид существует не для всех функциональных зависимостей.

Графическое описание является наиболее удобным для восприятия и анализа, так как в графике

сосредоточена значительная и очевидная информация о функции, например, интервалы выпуклости и вогнутости, точки перегиба и другие свойства функции, которые невозможно извлечь в процессе визуального анализа таблицы ее значений. Многовековая практика доказывает, что самыми эффективными являются графические модели, описывающие зависимость нескольких величин. Но графики можно построить только для зависимостей 3-х величин, т. е. для функций 2-х аргументов. Если же размерность пространства аргументов больше 2-х, то можно строить только графики срезов функциональных зависимостей, где заморожены (зафиксированы) все аргументы, кроме одного или двух. Но и с таким представлением значительно легче работать, чем визуально анализировать громоздкие таблицы или выполнять непростые математические вычисления с формулами аналитического представления сложных зависимостей, которые к тому же не всегда возможны.

Сказанное объясняет, почему авторы методик моделирования сложных систем и процессов разрабатывают графические модели, изображаемые в виде диаграмм (картинок), которые состоят из объектов (фигур) и различных связей между ними. Информативность такого представления многомерна, так как выражается в различных свойствах (цвет, фон, начертание и др.) и атрибутах (вес, размер, стоимость, время и т. д.) каждого объекта и связи.

### *Где применяются CASE-средства?*

Развитие диаграммной технологии моделирования привело к появлению CASE-средств (*Computer-Aided Software/Systems Engineering*), предназначенных для автоматизации различных операций с диаграммами, описывающими функционирование сложных систем, к которым относятся многие объекты экономики. CASE-средства помогают разобраться в сложной предметной области со слабо структурированными знаниями о ней, предоставляя возможность декомпозиции сложной системы на простые подсистемы и “дозированного” иерархического представления информации.

Разработке моделей перспективных технологий предприятия должен предшествовать глубокий анализ существующих технологий. Для этого необходимо построить их модели типа “как есть”. Построение таких моделей с использованием CASE-средств вынуждает системного аналитика глубоко разобраться в моделируемой системе, связать воедино разрозненные и неполные данные о ней. При этом CASE-средства играют роль “активизаторов”, заставляющих аналитиков ис-

кать недостающую информацию, разбираться в неточностях и противоречиях. Определенную пользу приносит уже сбор исходных данных для построения моделей. Ведь подготовить количественные характеристики невозможно без подробных знаний о предмете.

Разработанные с использованием CASE-средств модели деятельности создают формальную основу для проведения различного рода оценок. Механизмы оценок, как правило, встроены в CASE-средства. В первую очередь оцениваются затраты на выполнение отдельных работ и деятельности предприятия в целом. Сравнение различных вариантов выполнения работ позволяет выбрать наименее затратные из них.

Центральной частью CASE-средств является репозиторий, хранящий информацию о системе на всех этапах ее жизненного цикла и обеспечивающий согласованную работу всех участников разработки, создания и сопровождения системы. Для доступа к репозиторию и управления им используется специальное средство (навигатор по объектам репозитория), позволяющее просматривать и модифицировать объекты, хранящиеся в репозитории, а также осуществлять административные функции: удаление, управление доступом, экспорт, импорт и т. п.

Репозиторий включает не только объекты различных типов, но и отношения между ними, а также правила их использования и обработки. Все отчеты строятся автоматически по репозиторию. Функции управления и контроля также реализуются на основе репозитория. В частности, через репозитории может осуществляться контроль безопасности (ограничения доступа, привилегии доступа), контроль версий, контроль изменений и др.

Широкие возможности CASE-систем позволяют пользователям целиком сосредоточиться на собственно моделировании, не отвлекаясь на решение второстепенных вопросов, связанных с размещением элементов диаграмм, их компоновкой и т.п.

### *Для чего используются модели?*

Практика разработки сложных систем подтверждает концентрацию сложности на начальных этапах разработки (анализ требований) при относительно невысокой сложности и трудоемкости последующих этапов. На этапе анализа требований приходит понимание того, что будет делать будущая система и каким образом она будет работать, чтобы удовлетворить предъявляемые к ней требования. Нечеткость и неполнота системных требований, нерешенные вопросы и ошибки, допу-

щенные на этапах анализа и проектирования, порождают на последующих этапах трудные, часто неразрешимые проблемы и, в конечном счете, приводят к неуспеху всей работы в целом.

Построенные модели являются результатом начальных этапов разработки системы и техническим заданием на последующие этапы. Но они являются и самостоятельным (отделяемым) результатом, имеющим большое практическое значение. В частности модель “как есть” описывает существующие технологии на предприятии. Анализ этой модели позволит выявить узкие места и предложить рекомендации по их улучшению (независимо от того, предполагается на данном этапе автоматизация предприятия или нет).

Лучше один раз увидеть, чем сто раз услышать, поэтому модели в виде диаграмм являются прекрасным наглядным пособием для новых работников и позволяют быстро обучить их конкретному направлению деятельности предприятия. С помощью моделей можно также осуществлять предварительное рассмотрение новых направлений деятельности предприятия.

Комплексная модель системы и ее процессов является результатом этапа разработки системного проекта (анализа требований к системе), на котором требования заказчика уточняются, формализуются и документируются. Фактически на этом этапе дается ответ на вопрос: “Что должна делать будущая система?”. Выявление и согласование требований заказчика новой системы приводит к пониманию того, что же в действительности необходимо сделать. За этим следует проектирование или выбор готовой системы так, чтобы она в итоге как можно в большей степени удовлетворяла требованиям заказчика. Именно на этапе анализа требований закладывается фундамент успеха всего проекта автоматизации предприятия. В практике автоматизации сложных систем известно немало примеров неудачной реализации именно из-за неполноты и нечеткости определения системных требований.

Системный проект позволяет следующее: описать, “увидеть” и скорректировать будущую систему до того, как она будет реализована физически; уменьшить затраты на разработку и внедрение системы; оценить разработку по времени и результатам; достичь взаимопонимания между всеми участниками работы (заказчиками, пользователями, разработчиками, программистами и т. д.); обеспечить требуемое качество разрабатываемой системы.

На этапе анализа требований необходимо не только понять, что предполагается сделать, но и задокументировать это на достаточно простом и понятном заказ-

чику языке. Модели требований (логические модели) системы, состоящие из множества взаимосвязанных диаграмм, текстов, таблиц и т. п., как раз и являются такими документами. Этот результат (выход) системного проекта независим и отделяем от его разработчиков. Он не требует сопровождения и может быть передан другим лицам с целью подготовки и проведения следующих этапов работы.

### ***Какие особенности у функционального и объектно-ориентированного моделирования?***

В любой динамической системе протекают те или иные процессы. В экономических системах и их подсистемах протекают бизнес-процессы и рабочие процессы. Причем последние по определению отличаются от бизнес-процессов только тем, что они не преследуют получение прибыли. В дальнейшем мы не будем различать рабочие и бизнес-процессы, а будем называть те и другие бизнес-процессами и использовать для обращения к ним аббревиатуру БП. Таким образом, говоря о динамике в экономической системе, мы говорим о БП, в ней происходящих.

Современные методы моделирования экономических систем ориентированы на БП. Это связано с осознанием необходимости реорганизации БП перед автоматизацией предприятия (“автоматизировать существующие БП все равно, что асфальтировать дороги, по которым коровы ходят на пастбище”). Модели БП являются основой для размышлений и совместной работы группы специалистов над перспективами реорганизации и развития предприятия.

В теории и практике используются различные подходы к отображению моделей БП. Кратко остановимся на функциональном и объектно-ориентированном подходе. В функциональном подходе главным структурообразующим элементом является функция (действие, операция), в объектно-ориентированном подходе – объект.

Функциональный подход в моделировании БП сводится к построению схемы БП в виде последовательности операций, с которыми связаны материальные и информационные объекты, используемые ресурсы, организационные единицы и многое другое. Достоинством функционального подхода является наглядность последовательности и логики операций в БП предприятия, а недостатком – некоторая субъективность детализации операций.

Объектно-ориентированный подход предполагает вначале выделение объектов, а затем определение тех действий, в которых они участвуют. При этом различают пассивные объекты (материалы, документы, оборудование), над которыми выполняются действия, и активные объекты (организационные единицы, конкретные исполнители, программное обеспечение), которые осуществляют действия. Такой подход позволяет более объективно выделить операции над объектами и решить задачу о целесообразности использования этих объектов. Недостаток объектно-ориентированного подхода заключается в меньшей наглядности конкретных БП.

Особо отметим унифицированный язык моделирования *UML (Unified Modeling Language)*, т. е. стандартное средство описания систем, в основном проектов по созданию программного обеспечения. UML основан на объектно-ориентированном подходе в анализе, проектировании и программировании. К моделям в UML относятся, например, диаграммы классов, которые с концептуальной точки зрения описывают понятия изучаемой предметной области и увязывают эти понятия в строгую систему (каждый класс можно трактовать как некоторое сложное понятие моделируемой предметной области). Диаграмма классов определяет типы объектов системы и различного рода статические связи между ними.

В качестве объектов предметной области могут рассматриваться конкретные предметы или реальные сущности, например, клиент, заказ, изделия и т. п. Каждый объект характеризуется своим состоянием (точнее набором атрибутов, значения которых определяют состояние), а также набором операций для проверки и изменения этого состояния. Каждый объект является представителем некоторого класса однотипных объектов, определяющего их общие свойства. Все представители (экземпляры) одного и того же класса имеют один и тот же набор операций и могут реагировать на одни и те же сообщения.

Объекты и классы организуются с использованием принципов инкапсуляции, наследования и полиморфизма. Принцип инкапсуляции (упрятия информации) декларирует запрещение любого доступа к атрибутам объекта, кроме как через его операции. В соответствии с этим внутренняя структура объекта скрыта от пользователя, а любое его действие инициируется внешним сообщением, вызывающим выполнение соответствующей операции. Принцип наследования декларирует создание новых классов от общего к частному. Такие новые классы сохраняют все свойства

классов-родителей и при этом содержат дополнительные атрибуты и операции, характеризующие их специфику. Принцип полиморфизма декларирует возможность работы с объектом без информации о конкретном классе, экземпляром которого он является. Каждый объект может выбирать операцию на основании типов данных, принимаемых в сообщении, т. е. реагировать индивидуально на это (одно и то же для различных объектов) сообщение.

Таким образом, объектно-ориентированный подход заключается в представлении моделируемой системы в виде совокупности классов и объектов предметной области. При этом иерархический характер сложной системы отражается с использованием иерархии классов, а ее функционирование рассматривается как взаимодействие объектов. Жизненный цикл такого подхода содержит этапы анализа требований, проектирования и эволюции.

### ***Почему выбирают АРИС-моделирование?***

Следует заметить, что в 80-х годах прошлого столетия из-за обилия методов, языков, нотаций и процессов моделирования возникло предложение разработать единые стандарты. Многие выступили против этого предложения. Разработчики АРИС-концепции решили объединить (интегрировать) известные и хорошо проявившие себя методологии, установив между ними взаимосвязи и исключив избыточность (повторяемость). При этом они получили возможность с разных точек зрения и сторон описывать БП. Понятно, что чем больше точек зрения и сторон описания, тем точнее модель БП.

Для людей характерно многообразие точек зрения и представлений даже по существу одного и того же предмета или понятия. Поэтому, чтобы понимать друг друга, надо видеть или уметь показать связи между разными точками зрения и разными представлениями. Взаимопонимания следует добиваться не столько путем жесткой унификации и единообразия, сколько с помощью налаживания разветвленной и гибкой сети “мостов” и “дорог”, позволяющих легко ориентироваться в сложной предметной области. В этой связи следует отметить, что АРИС-методология помогает передавать идеи от человека к человеку и устанавливать взаимопонимание между людьми, причастными к построению моделей и их использованию. Она не навязывает одну точку зрения, язык, терминологию, форму записи (синтаксис) или семантику, а напротив, помогает выяв-



лять связи, переходить к разным точкам зрения, языкам, терминам, синтаксисам и семантикам.

Таким образом, наибольшую перспективу имеют комплексные (интегрированные) методологии моделирования бизнес-систем и БП, к которым относится АРИС-методология, позволяющая в зависимости от целей выбирать соответствующие модели. Аббревиатура АРИС расшифровывается как АРхитектура интегрированных Информационных Систем, т. е. АРИС архитектурно устроена как множество различных методов (от описания БП до реализации прикладных систем, автоматизирующих выполнение этих БП), интегрированных во взаимосвязанное целое с использованием современных информационных технологий.

Роль методологии заключается в регламентации основ разработки сложных систем. Она описывает подходы и последовательность шагов, позволяющих создавать высокоэффективные системы. Методология помогает охватить и учесть все важные этапы, шаги и моменты разработки. Более того, методология обеспечивает организационную поддержку, позволяющую скоординировать работу больших коллективов разработчиков.

АРИС-методологию используют для описания деятельности предприятия. С ее помощью можно задокументировать все БП, выявить в них недостатки, мешающие эффективной работе предприятия, получить на основе моделей БП нормативные документы для поддержки и автоматизации БП и т. д.

АРИС-диаграммы – это не только красивые картинки. Их можно применить с пользой, например, для “ручного” анализа, изучения, обсуждения и поиска путей улучшения БП. Когда картина бизнес-деятельности находится только в головах участников анализа, то выполнить такой анализ практически невозможно. Прежде чем анализировать, надо иметь предмет анализа, одинаково понимаемый всеми участниками процесса анализа. Поэтому АРИС-диаграммы часто рисуют в “Worde” с использованием его графических средств.

Архитектура АРИС (“дом АРИС”) – это совокупность технологий, которые обеспечивают разработку и совершенствование системы управления предприятием, а также проектирование и создание интегрированных информационных систем ее поддержки. АРИС-модели являются основой для анализа и оптимизации БП, для расчета стоимости БП, для подготовки к сертификации по международным стандартам качества, для документирования знаний фирмы, для выявления требований к корпоративной информационной системе (КИС), для разработки и внедрения самой КИС и для

достижения многих других целей в бизнесе.

### *Чем хороша программная среда ARIS 5.0?*

Здесь и в дальнейшем используется аббревиатура АРИС, когда мы рассматриваем или обращаемся к концепции или методологии, но когда мы имеем дело с программной средой, то используем аббревиатуру ARIS. Следует заметить, что АРИС отличается от ARIS не только по написанию, но и по существу. Например, в АРИС рассматривается пять видов моделей (организационные модели, функциональные модели, модели данных, модели входов-выходов и модели процессов или управления), а в ARIS можно построить всего четыре вида моделей (модели данных и входов-выходов объединены в один вид моделей).

Описывая статику и динамику экономической системы, строят множество диаграмм, которые классифицируются по видам, типам, уровням и срезам описания. Это позволяет с разных точек зрения взглянуть на моделируемую систему. Чем больше моделей можно построить, тем точнее (адекватнее) будет описана система и ее процессы. Это аналогично тому, как психологи изучают отношение человека к чему-либо, задавая ему десятки специально подобранных вопросов. Анализируя ответы на эти вопросы, они с той или иной степенью достоверности формируют мнение о человеке. Причем, чем больше вопросов, чем ближе они к теме, чем проще, взаимосвязаннее и ближе друг к другу эти вопросы, тем более достоверны выводы.

Аналогично построена АРИС-концепция и программная среда ARIS 5.0 для моделирования сложных систем и процессов. Во-первых, программная среда ARIS 4.0 позволяла строить 85 диаграмм, а ARIS 5.0 – уже 130. Во-вторых, эти диаграммы взаимосвязаны и информационно дополняют друг друга. В-третьих, каждая отдельная диаграмма достаточно проста, но их множество позволяет адекватно описывать сложные системы. В-четвертых, 20% диаграммных возможностей достаточно для моделирования систем и процессов примерно с 80%-й точностью (адекватностью). В-пятых, по каждой из диаграмм можно судить об одной из сторон системы, а по всем — о системе в целом.

Программную систему ARIS 5.0 можно использовать как CASE-средство, позволяющее построить до 130 типов диаграмм. Они используются, например, в качестве наглядной “двумерной” документации рассматриваемой системы и ее процессов. Создание и модификация этих диаграмм осуществляется с помо-

щью графического редактора (дизайнера). ARIS-дизайнер обеспечивает следующее: создание связанных диаграмм, в которых комбинируются графические, табличные, текстовые и другие объекты; создание и редактирование объектов в любом месте диаграммы; создание, перемещение и выравнивание групп объектов, изменение их размеров, масштабирование; сохранение связей между объектами при их перемещении и изменении размеров; автоматический контроль ошибок и др. Например, контроль синтаксиса диаграмм осуществляется при вводе и редактировании их объектов. Далее возможен контроль полноты и состоятельности диаграмм, где проверяется, например, все ли объекты диаграмм идентифицированы и отражены в репозитории.

Для создания и сопровождения системы управления предприятием с использованием ARIS осуществляется следующее: классификация БП; создание полных и согласованных моделей БП и их автоматизированная поддержка в актуальном состоянии; автоматическое документирование БП; автоматизированная поддержка создания и сопровождения документации по системе управления качеством; использование единой информационной системы для выполнения проектов по реинжинирингу БП, проектированию КИС, функционально-стоимостному анализу, имитационному моделированию и др.; автоматизированная поддержка процесса перехода от менеджмента качества к глобальному управлению качеством; создание и управление корпоративными знаниями и опытом.

ARIS 5.0 содержит инструменты моделирования е-процессов, т. е. электронных БП, что позволяет осуществлять переход от классического к е-бизнесу, а за ним к мобильному бизнесу (m-бизнесу). С примерами е-процессов пользователи компьютеров встречаются регулярно и охотно переходят на них. Например, многие уже забыли классический процесс “профилактика компьютерных болезней”, в котором пользователь регулярно проверял свои носители информации на наличие “компьютерных вирусов” и при их обнаружении “лихорадочно” искал “антивирусы”, ставил и запускал их на своем компьютере, если не достигал результата, то искал другие средства и т. д. Сейчас (при наличии соответствующего программного обеспечения) пользователь только настраивает параметры е-процесса “профилактика компьютерных болезней”, например, включает режим регулярной проверки своих носителей информации на “вирусы” и ежедневного (в обеденное время) пополнения своей базы “антивирусов” и базы знаний о состоянии профилактических дел. Пользова-

телю остается только анализировать этот е-процесс и изредка вмешиваться в него, например, для изменения настроек.

Безусловно, будущее за е- и m-процессами. Сейчас наблюдается бурное развитие е-бизнеса. С усовершенствованием и удешевлением мобильных средств связи е-процессы превратятся в m-процессы. Для моделирования, анализа и перехода от классических БП к е-БП, а от них к m-БП неоценимую помощь окажет программная система ARIS.

### *Что изложено в этой книге?*

Мы и наши коллеги имеем достаточный опыт использования АРИС и ARIS для формального описания роботизированных технологических комплексов и других сложных управляемых технических систем как средств автоматизации производственных процессов. Но мы отказались от соблазна теоретического описания и демонстрации практического использования АРИС в моделировании управляемых технических систем и их процессов, т. к. это заслуживает рассмотрения в отдельной работе.

В этой книге рассмотрены методы моделирования экономических систем и их БП. В качестве примеров использования изложенных методов построены более 100 статических и динамических моделей экономических подсистем и БП. Описана роль и функции информационных технологий в экономике. Изложены области практического применения бизнес-моделей. Основной упор сделан на БП, бизнес-информатику, АРИС-методологию и программную систему ARIS 5.0.

В 1-й главе введены в обращение основные понятия экономических систем и процессов. Особое внимание уделено производственным подсистемам экономики и производственным процессам. Для финансовой подсистемы даны определения основных понятий и приведен пример арбитражной стратегии. Все это сделано главным образом для того, чтобы на простых примерах продемонстрировать основные диаграммы и технику их построения еще до систематического описания методологии моделирования и построения ARIS-диаграмм. Например, на основе вербального описания арбитражной стратегии построена модель типа eEPC соответствующего БП.

Во 2-й главе рассмотрены основные понятия бизнес-информатики. Изложены особенности экономической информации. Описана роль и функции информационных технологий в экономических системах и процессах. Основным понятиям даны очень краткие (насколь-

ко это возможно) определения. Отдельный параграф посвящен конфиденциальной бизнес-информации и ее защите, где наряду с вербальным описанием представлена модель этой предметной области с соответствующей детализацией объектов этой модели.

В 3-й главе особое внимание уделено использованию экономической информации в управлении БП. Обоснована важность информационного описания БП и создания интегрированных информационных систем в экономике. Изложены основы процессной методологии и мотивирована необходимость перехода на процессно-ориентированное управление в экономических системах. В последних разделах главы рассмотрены некоторые аспекты инжиниринга и реинжиниринга БП.

В 4-й главе даны основы процессов обследования предприятий и их словесного (текстового, вербального) описания. Для моделирования, т. е. формального описания динамической системы и ее процессов, необходимо как минимум либо быть участником этих процессов, либо наблюдать за ними, либо иметь их текстовое описание. Лучше всего моделирует наблюдательный участник процессов при наличии их хорошего (понятного, непротиворечивого и т. д.) текстового описания. В этой главе обсуждаются методические особенности такого описания.

В 5-й главе изложена АРИС-методология применительно к бизнес-системам и БП. Здесь описаны принципы классификации моделей и области практического применения бизнес-моделей. Подробно описаны этапы бизнес-моделирования, виды и типы моделей, уровни моделирования, степени детализации моделей и многое другое. Последний раздел посвящен описанию управления корпоративными знаниями.

В 6-й главе описаны основы построения и использования программной среды (системы) ARIS версии 5.0. Здесь приведены многочисленные и оригинальные примеры ARIS-моделей и подробно изложена методика их построения. Технология работы в ARIS изложена не только вербально, но и формально (диаграммно), т. е. с использованием самой ARIS. Кроме множества примеров и ряда методологических тонкостей моделирования бизнес-систем и БП, изложена методика выполнения семантического контроля моделей.

В 7-й главе практически в полном объеме описаны возможности программной системы ARIS 5.0 с простейшим методологическим фильтром, который предоставляет пользователю минимальные возможности, например, позволяет построить всего восемь типов моделей вместо 130-ти в общем случае. Отдельный раздел посвящен формированию отчетов по моделям и

выполнению их элементарного анализа. В конце главы приведены ARIS-программы, предназначенные для анализа моделей. Эти программы достаточно документированы и занимают от одной до трех страниц, т. е. могут послужить поучительными примерами программирования в ARIS.

В 8-й главе приведены примеры ARIS-моделирования различных систем и процессов. Начиная с простых моделей (небольшая гостиница, интернет-салон, телевизионные игры) и заканчивая фрагментами сложных моделей (управление факультетом, деятельность научно-производственной фирмы), мы стремились продемонстрировать в первую очередь наглядность их диаграммного описания и только потом их практическое использование.

В 9-й (последней) главе построена математическая модель производственно-сбытовой фирмы и выведены формулы для вычисления выручки, объемов производства, цены продукции для случая прибыльного и бесприбыльного финансово-обеспеченного производства. Здесь описана имитационная динамическая ARIS-модель финансово-обеспеченного производства и представлен результат ее “прокрутки” (имитационного моделирования). В заключении описан алгоритм имитационной модели производственно-сбытовой деятельности малого предприятия, выпускающего мужские сорочки. В качестве информационной базы этого алгоритма использованы различные ARIS-диаграммы.

### *Кому рекомендуется эта книга?*

Многие дистрибьюторы CASE-средств организуют обучение моделированию за несколько дней. Но за 3-4 дня можно освоить лишь основы конкретного программного продукта. Для освоения методологии моделирования, анализа и проектирования требуются месяцы, в течение которых желательно прослушать, по крайней мере, семестровый университетский курс и принять участие хотя бы в одном проекте. Для тех, кто решил серьезно пройти этот нелегкий путь, и написана настоящая книга. Однако не только начинающие, но и специалисты в области моделирования сложных систем и процессов найдут здесь для себя много полезного и интересного.

Особенно полезной книга будет тем, кто применяет CASE-средства для моделирования, анализа, оптимизации и автоматизации БП и, в частности, использует для этого АРИС-методологию и одноименную программную систему. Книгу могут использовать менеджеры по информационным технологиям, преподаватели,

студенты и аспиранты, занимающиеся информатикой, бизнес-информатикой и родственными предметными областями. Она будет полезна всем, кто изучает управление бизнесом, как учебное пособие по теории бизнеса, ориентированного на информационные технологии, или как учебное пособие по теории бизнес-информатики. В частности, мы и наши коллеги используем материалы этой книги в учебном процессе по специальности “Прикладная информатика в экономике” в следующих дисциплинах: “Введение в специальность”, “Проектирование информационных систем”, “Моделирование экономических систем и процессов”, “Математические методы в экономике”, “Теория экономических информационных систем”, “Высокоуровневые методы информатики и программирования”, “Информационные технологии”, “Имитационное моделирование экономических процессов”, “Предметно-ориентированные экономические информационные системы” и “Анализ и оптимизация бизнес-процессов”.

Технику диаграммного описания целых предметных областей используют многие авторы монографий, учебных пособий и журнальных статей. Например, в книгах [9, 16] менеджмент и бухгалтерский учет изложены с использованием оригинальных диаграммных языков. К сожалению, эти языки мало кому известны и понятны, поэтому без помощи авторов таких книг их использование для обучения затруднительно и мало эффективно. Но если бы подобные книги писались с использованием общедоступного языка ARIS-диаграмм, то читатели, знающие ARIS, получили бы максимальный эффект от изучения таких книг. Наша книга предназначена в том числе и для тех, кто хочет писать книги с использованием ARIS-диаграмм и/или читать такие книги.

### ***Что мы использовали и кто нам помогал?***

При написании книги мы пользовались источниками [1] – [29] и учитывали мнение коллег, за что выражаем им свою благодарность.

Все диаграммы в книге построены с использованием программной системы ARIS 5.0, которая приобретена для учебного процесса кафедры “Системы управления и математического моделирования” (СУИММ) Южно-Уральского государственного университета (ЮУрГУ).

Московские фирмы Весть-МетаТехнология и ВИП Анатех регулярно проводят семинары, на которых наши коллеги и мы углубляем свои знания и приобретаем

новый бесценный опыт. Без участия в этих семинарах и использования информации, которой там всегда много, мы бы не смогли подготовить материалы этой книги. Поэтому выражаем свою признательность сотрудникам указанных фирм и всем тем, кто популяризирует в России современные CASE-технологии, АРИС и ARIS как очень нужную и полезную для науки, образования и практики работу.

В некоторых местах этой книги мы использовали фрагменты лучших курсовых и дипломных проектов студентов кафедры СУИММ ЮУрГУ. Этих студентов мы уже отблагодарили, оценив их труд на “хорошо” и “отлично”. Особо отметим аспиранта, ассистента и студента, для которых часть материалов этой книги по праву является их первой публикацией.

В подготовке материала 7-й главы участвовал аспирант кафедры СУИММ Долгих А. А. Он также написал раздел 8.6, содержание которого частично отражает следующую тему его научных изысканий “Разработка методики системного анализа опережающих информационных технологий и управления процессами их эффективного освоения и промышленного внедрения”.

Раздел 6.9 вместе с нами писала ассистент кафедры СУИММ Зайнашева Т. Г. Мы надеемся, что она продолжит освоение и внедрение в учебный процесс кафедры АРИС-методологию и программную систему ARIS. Желаем ей успехов в этом.

Наш одаренный студент Михайлов Александр принимал участие в написании разделов 7.5 - 7.8. Как хороший писатель он пока не состоялся, но информатик и программист – уже хороший. Заранее просим воздержаться маститых программистов от критики приведенных в этой книге пяти программ. Уверяем Вас, что Александр справился с поставленной задачей хорошо, т. е. написал максимально короткие программы анализа ARIS-моделей для каждого из 4-х возможных видов. Эти программы, конечно, не столь универсальны, как известные ARIS-скрипты, но они на порядок (в десятки раз) короче и поэтому у нас появилась возможность включить их в книгу, а у Вас, дорогие читатели, впервые появилась возможность видеть и анализировать тексты ARIS-скриптов (программ) в отечественной литературе. За это мы Александра сердечно благодарим.

Мы заранее благодарим пока что безымянного читателя этой книги и особенно наших коллег – преподавателей, за то, что они внимательно прочтут наш труд, найдут в нем недостатки и, возможно, ошибки и обязательно сообщат нам по электронному адресу [Telegin@miass.tu-chel.ac.ru](mailto:Telegin@miass.tu-chel.ac.ru). Мы будем также признательны тем, кто подскажет нам пути улучшения текста



этой книги, новые примеры моделей (желательно короткие) и т. д. Если таких читателей будет много, то мы обязательно переиздадим эту книгу в исправленном и дополненном виде, где сошлемся на тех, кто поможет нам улучшить ее содержание.

Большую помощь в улучшении стиля изложения всего материала нам оказала литературный редактор

книги, кандидат филологических наук Коваль М. Л., за что мы ее сердечно благодарим. Выражаем искреннюю благодарность Максимову В. И. за помощь в оформлении этой книги. Фотографии упомянутых здесь наших помощников можно увидеть в разделе 9.5. Там они “проходят” как сотрудники малого предприятия по производству мужских сорочек.

***Миасс, февраль 2002 г., Авторы***

# 1. МОДЕЛИРОВАНИЕ ОСНОВ ЭКОНОМИЧЕСКИХ СИСТЕМ И ПРОЦЕССОВ

Для изложения теории моделирования экономических систем и процессов рассмотрим основные понятия экономической системы, ее подсистем и экономических процессов. При этом для большинства вводимых понятий составим ARIS-диаграммы, моделирующие статические или динамические взаимосвязи между ними и позволяющие глубже понять, проще запомнить и эффективнее анализировать их взаимодействия. Систематическое изложение методологии моделирования систем и процессов начнется с 5-й главы. Здесь же даны только минимально необходимые пояснения к построенным моделям.

**Модель** – это мысленно представленный или изображенный (например, нарисованный на бумаге), или изготовленный (например, бумажный макет) образ оригинала, который замещает оригинал и отражает наиболее важные черты и свойства оригинала. ARIS-модели – это изображенные диаграммы экономических систем и процессов, на основе которых можно изучать, анализировать, оптимизировать и управлять бизнес-процессами.

Модель, представляющая собой совокупность математических соотношений, называется математической. Основам математического моделирования бизнес-процессов производственно-сбытовой фирмы посвящена последняя глава книги.

## 1.1. Системы и их классификация

**Система** – это множество взаимосвязанных объектов, которые совместно реализуют определенные цели и имеют свойства, отсутствующие у отдельных объектов. **Объекты** – это составляющие части системы, причем, система имеет конечное число объектов. **Свойства** – качества объектов, дающие возможность количественного описания системы в определенных величинах. **Связи** – это то, что соединяет объекты и свойства системы в единое целое.

В окружающем нас мире все системно. Системами являются ансамбль атомов и молекул, авиалайнер, коллектив фирмы, население города. Транспортные сети, жилые кварталы, библиотеки также образуют системы. Системами являются ситуации, проблемы, явления. В системах может быть неизвестно точное количество объектов. Могут быть неизвестны свойства объектов и их связи. Например, часто неизвестны движущие силы общества, политических партий, движений, сообществ и т. п.

Несмотря на большое многообразие систем, они имеют много общих черт и особенностей. Для того, чтобы разобраться в многообразии систем, их необходимо классифицировать, т. е. разделить, а затем объединить по определенным признакам. Существует много способов классификации. На рис. 1.1.1 представлена ARIS-диаграмма, классифицирующая системы наиболее распространенным способом.

Забегая вперед, заметим, что в программной среде

ARIS 5.0 можно построить четыре вида (представления) моделей. Для каждого вида (в зависимости от используемого методологического фильтра) можно построить от одного до нескольких десятков типов моделей (в ARIS 5.0 с полным фильтром допустимы 130 типов диаграмм, а с упрощенным фильтром – всего 8). Вид модели на рис. 1.1.1 называется “Представление данных” (Data View), а тип “Модель технических терминов” (Technical terms model). Здесь использованы два объекта – “Технический термин” (прямоугольник с символами FB в правом нижнем углу) и “Кластер” (прямоугольник с боковыми сторонами из тройных линий). Связь, проводимая от технического термина к кластеру, имеет тип “изображает, описывает (depicts)”, а между техническими терминами – “имеет связь с (has relation with)”.

Заметим, что ARIS-диаграмма – это тоже система, у которой объекты имеют свойства и атрибуты, сосредоточенные в таблицах. **Атрибут** – это признак, характеризующий объект. Например, атрибутами выводимых на экран символов являются шрифт, цвет, размер и т. п. В таблице атрибут – это имя одного и того же поля всех *кортежей*. Содержимое заполненных или численных атрибутов можно показать на диаграмме или в ARIS-отчете, который при помощи *мастера отчетов* автоматически генерируется на основе разработанной модели. **Мастер** – это специальное приложение (или компонент приложения как в ARIS), которое упрощает использование или настройку системы. Например, “Мастер установки принтеров” упрощает процедуру установки принтера. Мастер обычно реализует

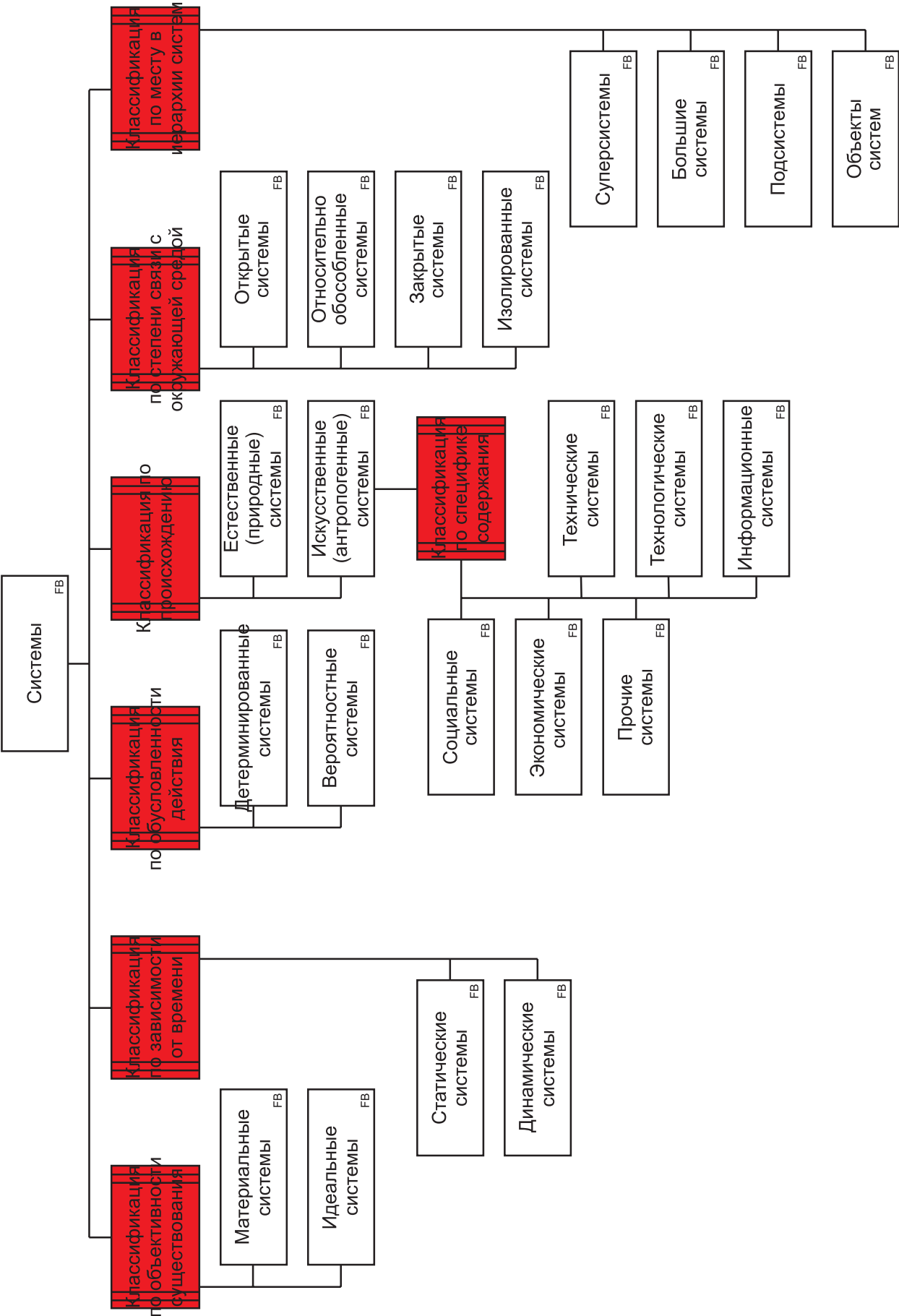


Рис.1.1.1 Классификация систем

ся в виде последовательности диалоговых окон, которые содержат пояснения и простые вопросы и позволяют пользователю шаг за шагом выполнить довольно сложную процедуру. В следующих разделах мы подробно рассмотрим возможности составления отчетов в системе ARIS 5.0, а сейчас дадим пояснения к рис.1.1.1.

По происхождению различают системы: естественные (природные), например, звездные образования, Солнечная система, планеты, материки, океаны и т. п.; искусственные (антропогенные), т. е. обязанные своим происхождением труду человека или побочным результатам труда (например: города, предприятия, машины). Из последних особенно отметим такие системы, как народное хозяйство страны, отрасль, регион, предприятие, цех, участок и т. п. По объективности существования системы могут быть материальными (существующими объективно, т. е. независимо от сознания человека); идеальными, т. е. “сконструированными” в сознании человека в виде гипотез, образов, представлений, ARIS-моделей. В последнем случае они могут быть в виде диаграмм, графиков, формул, уравнений, знаковых схем, музыкальных и зрительных образов и т. п. Различают суперсистемы (мировая экономика, Вселенная, космос); большие системы (национальная экономика); подсистемы (экономика региона, отрасли); объекты систем (предприятие, фирма).

Системы могут быть классифицированы другими способами в зависимости от целей, условий и поставленных задач.

## 1.2. Экономические системы

Предметом нашего рассмотрения являются экономические системы, которые характеризуются определенными классификационными признаками. Это системы искусственные; материальные; открытые; динамические; вероятностные; на различных уровнях это суперсистемы, большие системы, подсистемы или их объекты.

Искусственность экономических систем означает, что они созданы трудом человека, даже если максимально используют природный ресурс. Искусственность предполагает также большую степень возможного разнообразия систем, что и обуславливает многообразие экономик.

Материальный характер экономических систем означает не только объективность их существования, но и тот или иной уровень материальных и финансовых затрат. Для информационных подсистем экономики,

например, необходимы значительные затраты на покупку компьютерной техники и технологии.

Экономические системы являются системами открытого типа, так как покупка или продажа товара связана с открытостью рынка, открытостью деятельности фирмы. Однако при этом любая фирма борется с промышленным шпионажем, тщательно оберегает коммерческие и производственные секреты.

Экономические системы являются системами динамического типа, они подвержены старению, развитию, движению, прогрессу и регрессу, делению и слиянию и т. д. В любой динамической системе протекают те или иные процессы. Если эти процессы не совершенствоваться, то система деградирует, а если ими не заниматься (не поддерживать), то система прекратит свое существование. Желательно все процессы системы прогнозировать, предвидеть, влиять на их развитие.

Экономические системы характеризуются вероятностью структуры, функций, целей, задач, ресурсов и т. д. Это значительно повышает роль индивидуальных, творческих начал в управлении системами, роль учета тех факторов, которые поведение фирмы делает более предсказуемым. В детерминированных системах одной и той же причине всегда соответствует четкий, строгий, однозначный результат. В системах вероятностного типа одной и той же причине в одних и тех же условиях может соответствовать один из нескольких возможных результатов. Для их предсказания используется теория вероятностей. Примерами вероятностных систем являются группа студентов и цеховой персонал, которые приходят на учебу или работу (каждый раз число студентов и рабочих может быть различным, как и их состав).

Система, как правило, погружена в более сложную систему (надсистему) и содержит более простые системы (подсистемы). *Надсистема* – это окружающая систему среда, в которой функционирует система. *Подсистема* – это часть системы, реализующая цели, согласованные с целями системы (например, подсистема может осуществлять часть целей системы).

*Основная цель экономики* (как системы) – удовлетворение неограниченных потребностей человека товарами и услугами при ограниченных ресурсах. К объектам экономики относятся *промышленные предприятия, торговые фирмы, банки* и т. д. Различные объединения этих объектов образуют экономические подсистемы, например, производственную подсистему, финансово-кредитную подсистему и др. *Производственная подсистема* подразделяется на *отрасли материального производства – промышленность,*

сельское хозяйство, строительство и др. Финансово-кредитная подсистема – это множество государственных и коммерческих банков, их филиалов, страховых обществ, различных фондов и т. п. Надсистема экономики – природа и общество.

На рис. 1.2.1 изображен фрагмент ARIS-диаграммы подсистем экономической системы. Это диаграмма модели структурного типа (Structuring Model) процессного вида (Process View), в которой используется всего один объект (прямоугольник с закругленными углами), обозначающий “понятие предметной области”, и одна связь (ломаная линия со стрелкой) типа “содержит” (contains).

Для выполнения своей основной цели экономическая система использует ресурсы, которые подразделяются на *природные, трудовые (рабочая сила), производственные (средства производства), финансовые и информационные*. При этом в экономической системе происходят следующие процессы: изучается спрос на товары и услуги, формируются ресурсы, производится продукция, продаются товары (предметы потребления), предоставляются услуги и т. д.

На рис. 1.2.2 представлена укрупненная модель производственной подсистемы экономики, которая сама является подсистемой природы и общества. Очевидно, что по отношению к экономической системе человек выступает в роли потребителя и в роли производителя (работника), т. е. он находится и в надсистеме и в подсистемах. Вид модели на рис. 1.2.2 является процессным типа eEPC, в котором изображен поток двух функций и их входов-выходов. Функции (работы, операции, действия и т. д.) во всех процессных моделях изображаются прямоугольником с закругленными углами, внутри которого записано название функции. Друг с другом функции связываются штрихованной прямой или ломаной линией со стрелкой, показывающей направление функционального потока. Входы-выходы (общие ресурсы) изображаются прямоугольником с двойной границей (двойная рамка), внутри которого записано название входа-выхода. Для выполнения функции требуется один или несколько входов. В результате выполнения функции создается один или несколько выходов. Часто один ресурс является выходом для одной функции и входом для другой.

На рис. 1.2.3 приведена диаграмма организационного вида (Organizational View), на которой представлен фрагмент иерархической организационной структуры (тип модели – Organizational chart) национальной экономики России. Здесь единственный использованный объект называется “Организационная единица”, а тип

связи – “формируется из (is composed of)”.

### 1.3. Производственные ресурсы (средства производства)

Производственные ресурсы (средства производства) разделяются на *средства труда* и *предметы труда*. Средства (орудия) труда (станки, инструмент, машины, роботы, ЭВМ и т. п.) участвуют в нескольких производственных циклах вплоть до их замены вследствие морального или физического износа. Предметы труда (материалы, комплектующие изделия, энергоресурсы и т. п.) участвуют в одном производственном цикле. Средства труда “прикладываются” (как правило, при помощи трудовых ресурсов) к предметам труда, т. е. выполняется производственный или офисный процесс. В результате этого процесса получается изделие (товар) или предоставляется услуга. Например, автоматическая стиральная машина (средство труда), обрабатывая грязное белье (предмет труда), предоставляет услугу. При этом программу стирки задает оператор, т. е. человек как составляющая трудовых ресурсов. При более подробном анализе процесса стирки мы увидим, что в нем задействованы все экономические ресурсы (природные – вода, информационные – данные о температуре воды и времени стирки для конкретного вида белья, финансовые – зарплата оператора). Примером информационной услуги является сбор сведений о платежеспособности клиента банка, где средством труда является ЭВМ, а предметом труда – информация и ее носители.

Особое место среди средств производства занимает *освоенная земля* (земли, не вовлеченные в хозяйственный оборот, являются природным ресурсом). Основная функция земли – участвовать в производстве сельскохозяйственной продукции. Поскольку она многократно участвует в производственных циклах, то в этом смысле является средством труда, но поскольку ее обрабатывают (к земле прикладывают) орудия труда, то ее можно было бы отнести к предметам труда.

Накопленные *производственные средства* подразделяются на *основные производственные средства* (ОПС) и *оборотные средства* (ОС). ОПС состоят из накопленных средств труда (станков, машин, компьютеров и т. п.). ОС состоят из накопленных предметов труда (кожи, зерна, горюче-смазочных материалов и т. п.).

ОПС в течение длительного времени обслуживают процесс производства, сохраняя при этом свою натуральную форму и частично (в меру своего изнашивания)

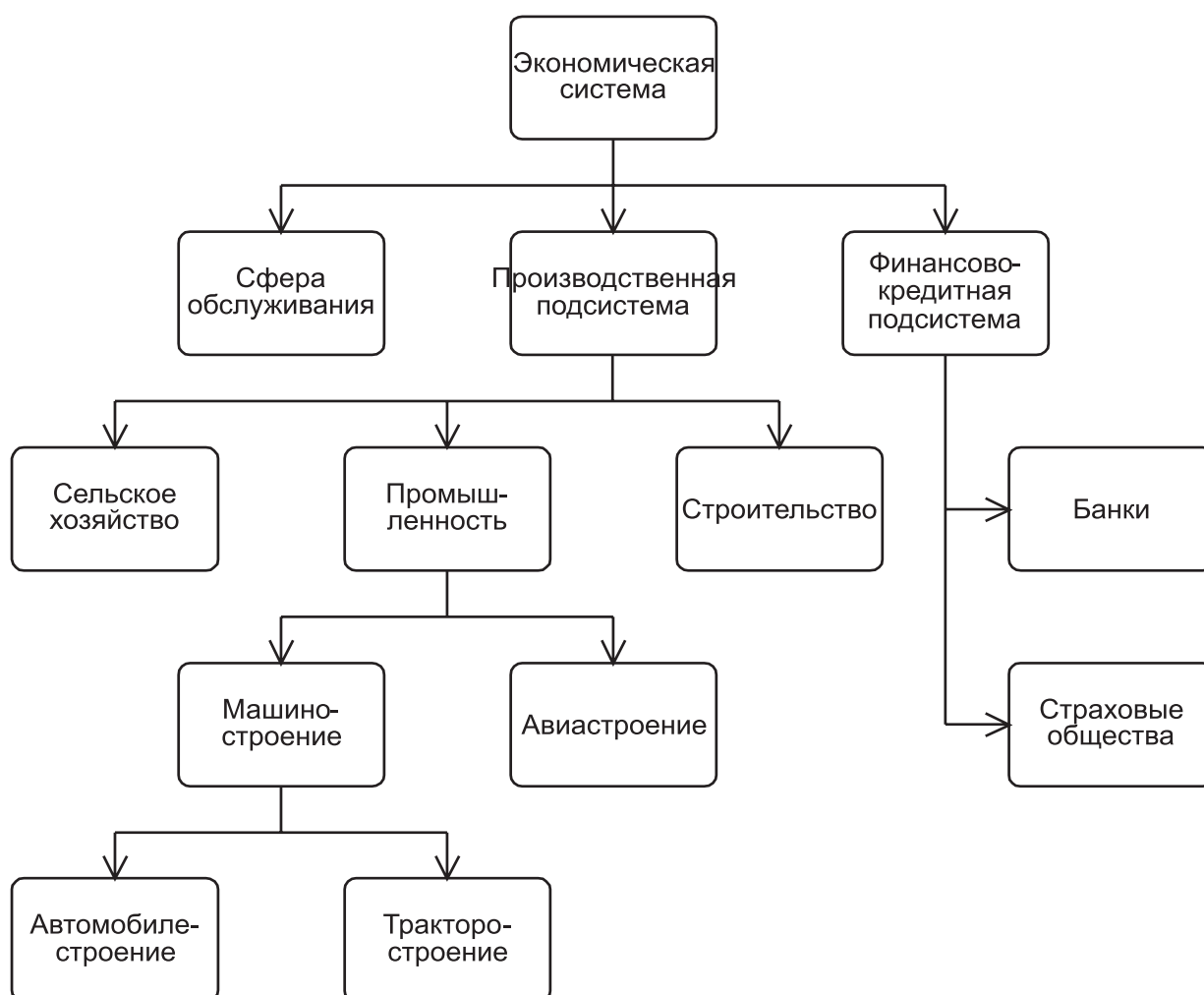


Рис.1.2.1 Фрагмент ARIS-диаграммы структурной модели подсистем экономической системы

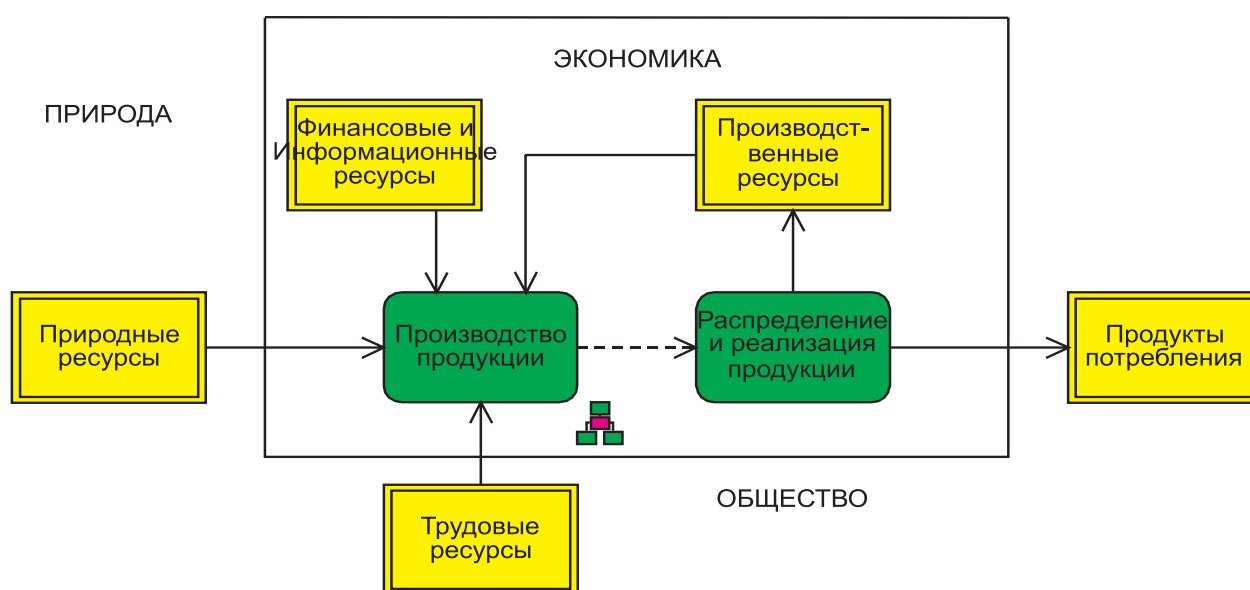


Рис.1.2.2 ARIS-диаграмма потока двух функций и входов-выходов в производственной подсистеме экономики

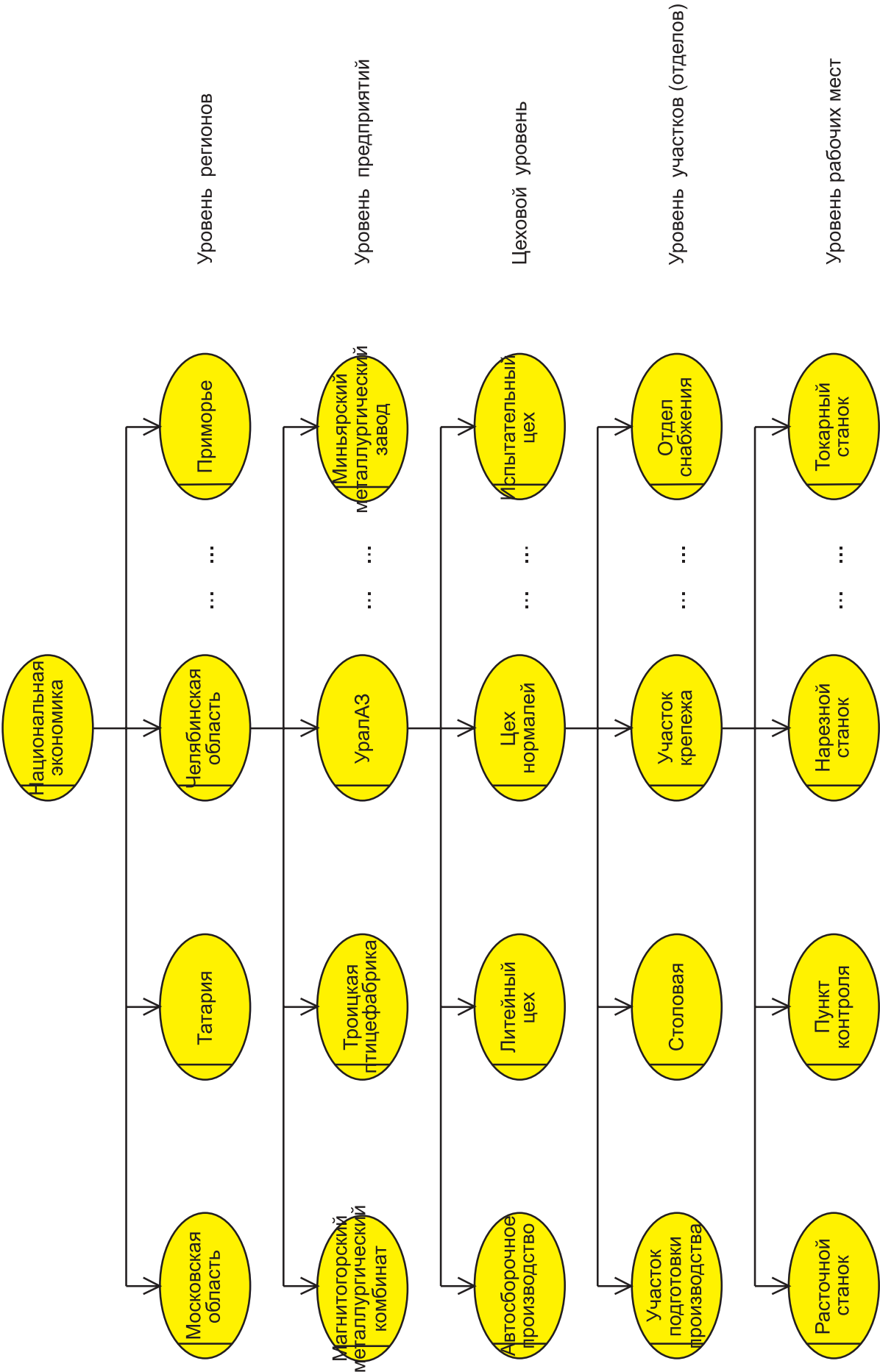


Рис.1.2.3 Фрагмент организационной диаграммы национальной экономики



участвуя в образовании стоимости производимого продукта. ОПС переносят свою стоимость в изделия частями и своей сущности не меняют (станок остается станком), но постепенно изнашиваются. Поэтому их необходимо восстанавливать или заменять. Восстановление (*простое воспроизводство*) ОПС осуществляется за счет *амортизационных отчислений*. Замена или увеличение ОПС (*расширенное воспроизводство*) осуществляется, в основном, за счет капитальных вложений.

ОС, т. е. накопленные предметы труда, подразделяются на *производственные запасы* и *незавершенное производство*. Незавершенное производство состоит из предметов труда, еще не перешедших в состояние готового изделия (товара), но находящихся в стадии изготовления. ОС переносят свою стоимость в изделия полностью и меняют свою сущность, например, кожа превращается в кожаную обувь, нефть – в бензин и т. п.

На рис.1.3.1 изображена диаграмма классов (UML Class diagram), связанных с понятием экономические ресурсы. Класс изображается прямоугольником с тройной нижней границей, внутри которого записано название класса. Каждый класс можно трактовать как некоторое понятие моделируемой предметной области. В диаграмме классов описывается не только иерархия понятий (как в структурной диаграмме), но и многое другое. Для описания иерархии используется тип связи “обобщение”, которая изображается в виде ломаной линии с большой треугольной стрелкой, входящей в обобщающее понятие. Каждый класс может иметь различные атрибуты (прямоугольник с символом “А” в правом нижнем углу) и операции (прямоугольник с символом “О” в правом верхнем углу). Класс в своих атрибутах хранит данные (понятия), а в операциях возможные действия (функции) над ними. Например, в сложное понятие (класс) “Трудовые ресурсы” входят понятия “Рабочие”, “Служащие” и т. д., которых можно “Воспитывать”, “Обучать” и т. д. Кроме обобщающей связи между двумя классами, могут устанавливаться “ассоциации”, под последними можно понимать действия, посредников или все, что ассоциируется с этими классами. Например, накопление предметов труда создает оборотные средства. Поэтому между классами “Предметы труда” и “Оборотные средства” установлена ассоциация “Накопление”, равно как и между классами “Средства труда” и “ОПС”.

## 1.4. Производственные процессы

В результате функционирования экономики за год все отрасли материального производства (промышленность, сельское хозяйство, строительство и др.) создают валовой внутренний продукт (ВВП). В натурально-вещественной форме ВВП распределяется на средства труда и предметы потребления, в стоимостной форме – на *фонд возмещения выбытия основных средств (амортизационный фонд)* и *вновь созданную стоимость (национальный доход)*.

В процессе создания ВВП производственная подсистема экономики производит и вновь потребляет промежуточный продукт. *Промежуточный продукт* – это предметы труда, используемые для *производственного потребления*.

В производственном процессе каждый природный ресурс претерпевает несколько преобразований (переделов) в промежуточную продукцию перед тем, как стать составной частью предмета потребления или средства труда. Промежуточная продукция – это топливо, энергия, сырье, материалы, комплектующие изделия и т. п. Для примера рассмотрим следующую цепочку переделов: рудное тело – добытая руда – чугун – сталь – металлопрокат – металлопродукция – продукция машиностроения. В этой цепочке рудное тело – это природный ресурс, руда – это сырье, чугун, сталь, металлопрокат – это материалы.

Следует обратить внимание на отсутствие абсолютно четкой грани между промежуточным продуктом и предметом потребления. Например, цемент, проданный населению, является предметом потребления, в то время как цемент, закупленный строительным предприятием, является промежуточным продуктом.

Каждая производственная подсистема (завод, фабрика, рудник, шахта, электростанция и т. п.) обладает средствами труда, которые позволяют выполнить один или несколько производственных процессов. Поступающие на вход процесса предметы труда перерабатываются в продукты труда, которые, в свою очередь, могут быть либо предметами труда в другом процессе, либо средствами труда, либо предметами потребления.

На рис.1.4.1 представлена диаграмма функционального потока и потока выходов основных видов деятельности производственно-сбытовой фирмы (предприятия), где показана последовательность и логика выполнения основных функций (работ) фирмы. Самый первый объект этой диаграммы является запускающим событием с названием “Начало непрерывного



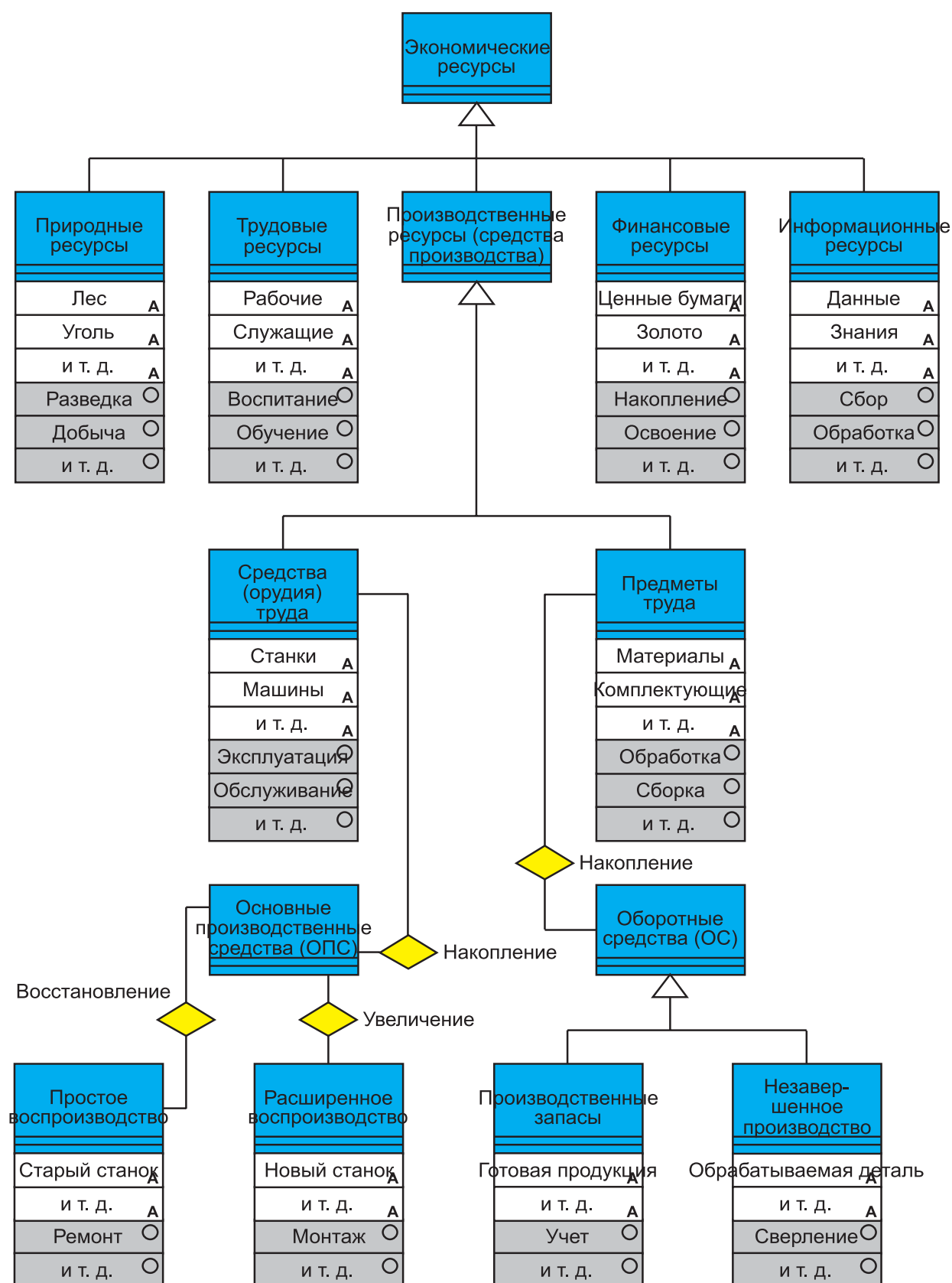
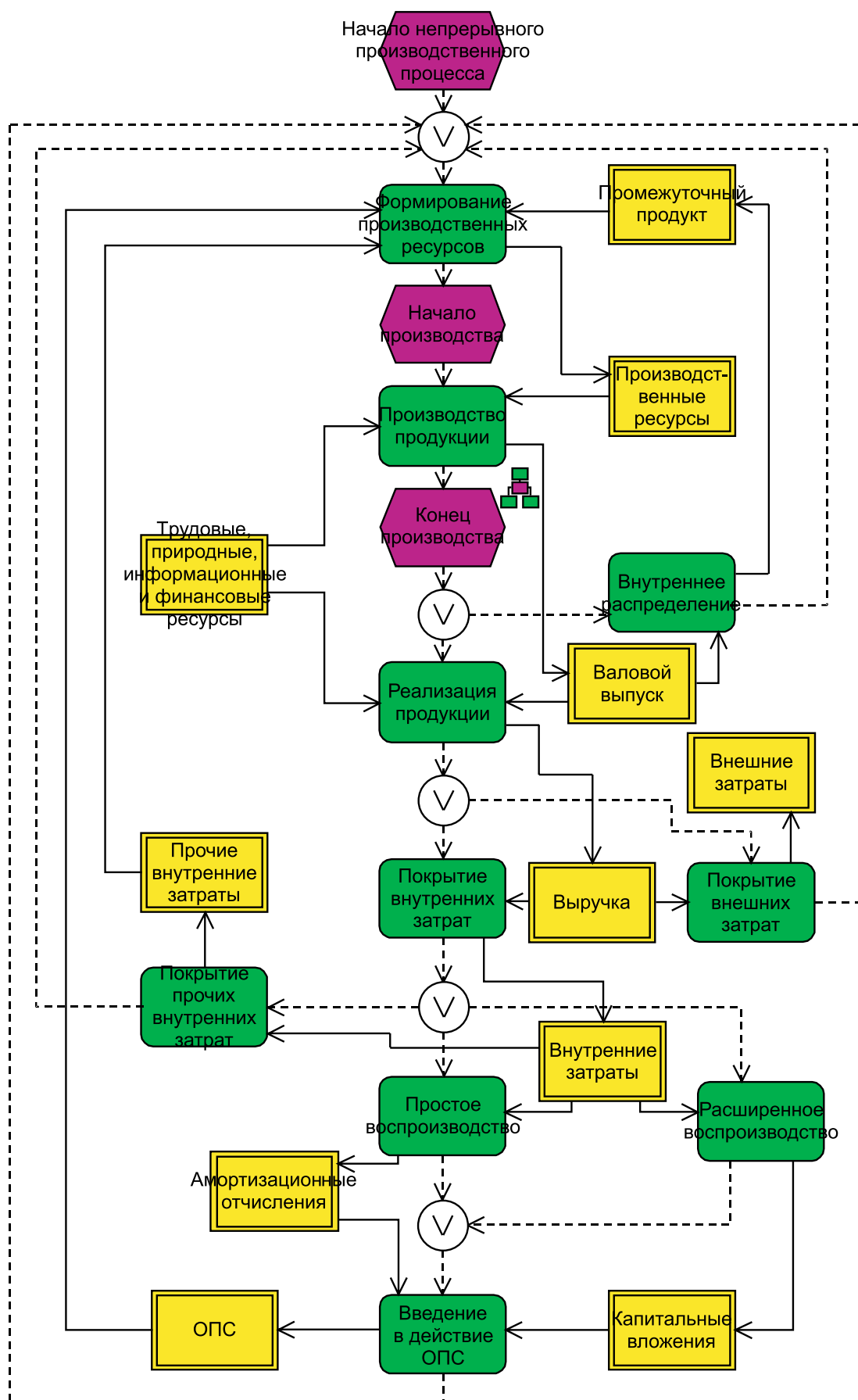


Рис.1.3.1 Диаграмма классов для понятия “Экономические ресурсы”



производственного процесса”, которое изображается шестиугольником. Каждая функция на диаграмме имеет одну входящую и выходящую штрихованную линию со стрелкой, которая указывает последовательность выполнения функций. Если после очередной функции могут выполняться одновременно две или более функций, то при помощи логического оператора одна штрихованная линия разделяется на две или более линий. Логический оператор используется также для объединения нескольких штрихованных линий в одну. Но основное назначение логического оператора – описание логики выполнения процесса. На рис. 1.4.1 использован логический оператор “или”, изображенный в виде символа “V” внутри круга. Каждая функция может иметь несколько сплошных линий входа и выхода, которые описывают поток ресурсов (входов) и результатов выполнения функции (выходов) рассматриваемого процесса.

В процессе на рис. 1.4.1 первой выполняется функция “Формирование производственных ресурсов”, так как без ее выхода “Производственные ресурсы” невозможно выполнить функцию “Производство продукции”. Полученный “Валовой выпуск” можно полностью реализовать или часть его оставить для внутреннего потребления. Поэтому за функцией “Производство продукции” следует логический оператор “или” с двумя выходами, т. к. вслед за этой функцией могут выполняться одновременно функции “Внутреннее распределение” и “Реализация продукции” или только одна из них. Однако без реализации продукции мы имели бы натуральное хозяйство, а вот без внутреннего (натурального) распределения валового выпуска работают многие фирмы. Самый первый логический оператор на рис. 1.4.1 имеет пять входов и один выход. В рассматриваемом процессе производственные ресурсы могут формироваться как за счет внутреннего распределения (если таковое имеется), так и вводом в действие ОПС и (или) покрытия прочих внутренних затрат.

Все функции на рис. 1.4.1 являются сложными, т. е. сами являются процессами, поэтому их желательно детализировать. На рис. 1.4.2 уточнена (детализирована) функция, а точнее БП “Производство продукции”. Обратите внимание на то, что на рис. 1.4.1, во-первых, перед функцией “Производство продукции” изображено событие “Начало производства”, во-вторых, в нижней правой части функции “Производство продукции” изображена пиктограмма детализации этой функции, в-третьих, за функцией “Производство продукции” следует событие “Конец производства”. Эти же два события на рис. 1.4.2 начинают и заканчивают БП “Производство продукции”.

Первая функция процесса на рис. 1.4.2 обеспечивает производство необходимыми ресурсами. Выходом второй функции “Изготовление продукции” является поток товаров (на рис. 1.4.2 не изображен). Если он достаточно близок к запланированному потоку, то считается, что цели достигнуты, т. е. производственная система работает без сбоев (идеально). Если же этот поток недопустимо отличается от запланированного, то информация об этом поступает по каналу обратной связи через событие “Цели не достигнуты” в функцию “Анализ отклонений от целей”, при выполнении которой исследуются причины появления отклонений. Результаты анализа используются для разработки системы мероприятий по устранению возникших отклонений, а также устранению причин, приведших к отклонениям. Вследствие принятых решений можно воздействовать как непосредственно на производственную систему (производственный персонал), так и на дополнительное ресурсное обеспечение, т. е. осуществлять либо дополнительное выделение ресурсов, либо их сокращение. В результате этих мероприятий стабилизируется деятельность производственной системы.

На рис. 1.4.2 впервые использован логический оператор “исключающее или” (символ X внутри круга). Этот оператор может иметь один вход и несколько выходов. Он работает следующим образом. Если вход (штрихованная линия со стрелкой) в операторе “исключающее или” активен, т. е. предшествующая функция выполнена, то только один из 2-х или нескольких возможных выходов может быть активным, что обеспечит запуск на выполнение следующих за оператором функций. Какой именно из выходов активен, зависит от наступившего переключающего события. На рис. 1.4.2 переключающими являются события “Цели достигнуты” и “Цели не достигнуты”.

В модели на рис. 1.4.2 использована обратная связь. Суть обратной связи заключается в том, что информация о выходе управляемой системы (выходном продукте, товаре, услуге) поступает в блоки принятия управленческих решений и используется в процессе управления.

На рис. 1.4.3 приведена диаграмма основных процессов конкретного металлургического предприятия, где отражены иерархические связи (сплошные линии со стрелкой) процессов и последовательность (штрихованные линии со стрелкой) выполнения этих процессов. Для процессов передела указаны входные ресурсы и выходная продукция, а также подразделения, в которых эти переделы осуществляются. Переделы изображаются двухмерной “стрелкой”, внутри которой

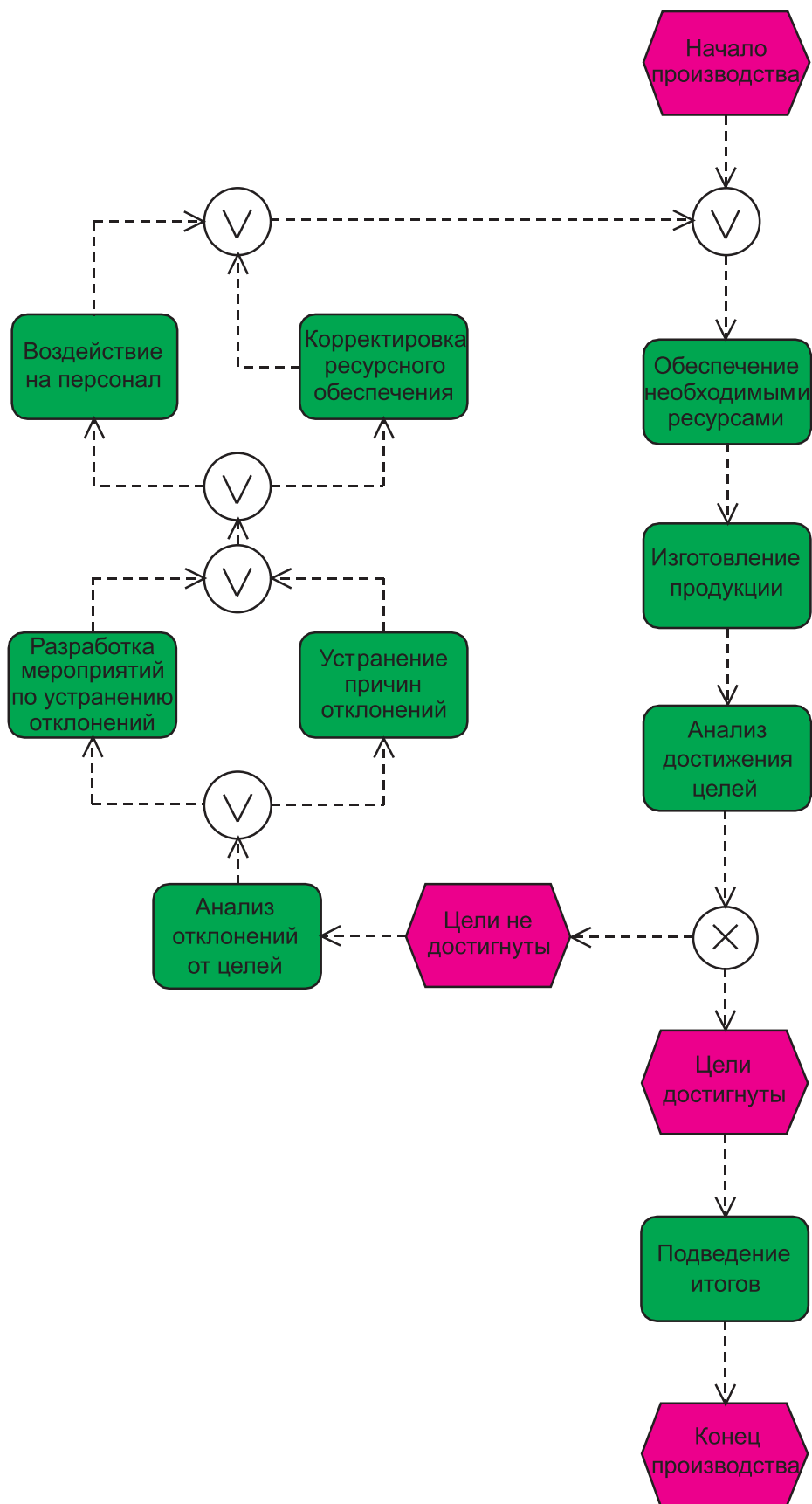


Рис.1.4.2 Функциональный поток, детализирующий БП “Производство продукции”

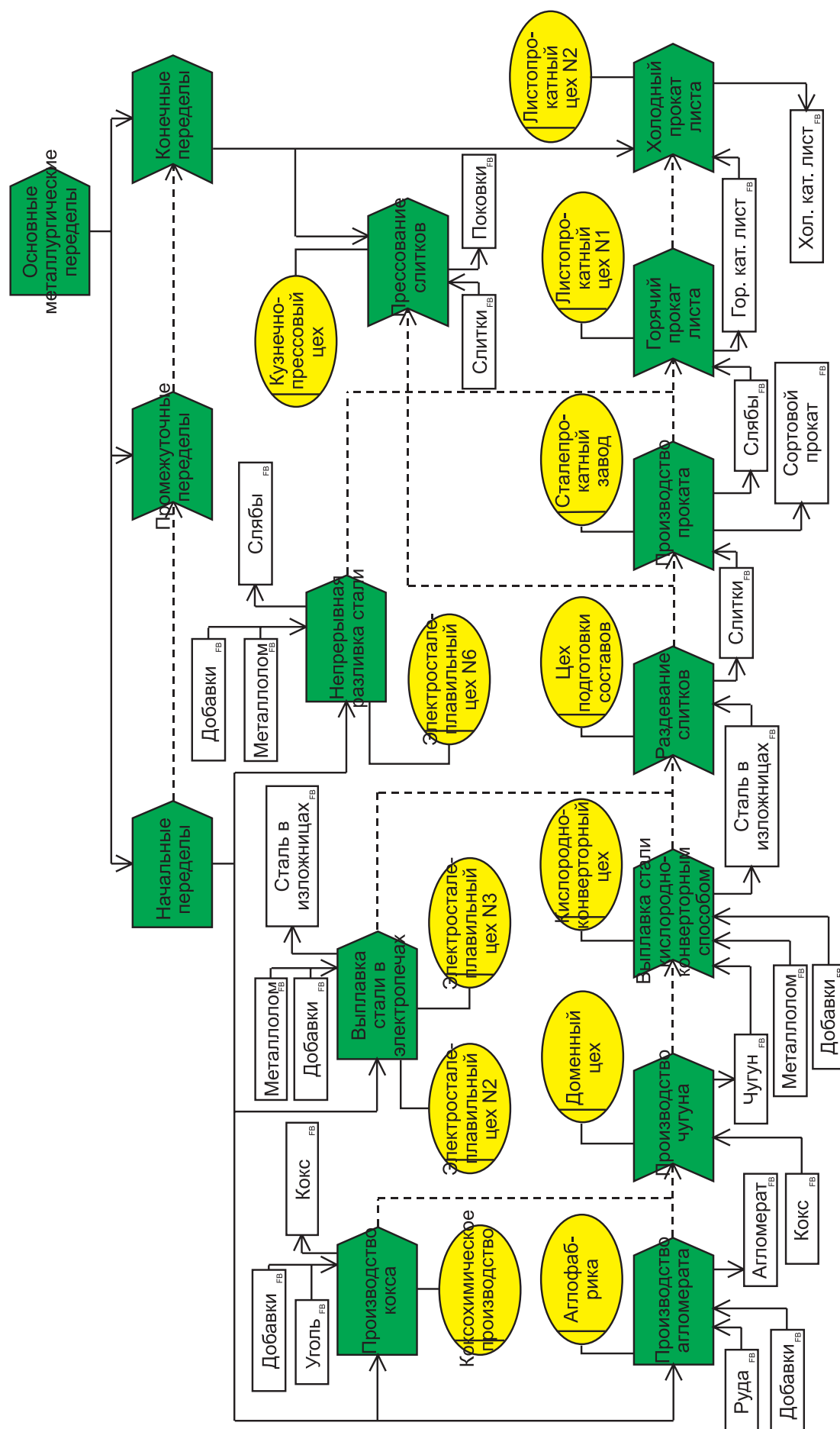


Рис.1.4.3 Диаграмма основных переделов металлургического производства

записывается название передела, например, “Производство чугуна”. Подразделение предприятия, в котором этот передел осуществляется, изображается перечеркнутым слева эллипсом. Для изображения входов и выходов переделов применяется объект “технический термин” (прямоугольник с символом FB в правом нижнем углу).

Из диаграммы на рис. 1.4.3 видно, что “Основные металлургические переделы” моделируемого предприятия представляются в виде последовательности трех комплексных процессов: “Начальные переделы”, “Промежуточные переделы”, “Конечные переделы”. К первому из них относятся процессы “Производство кокса”, “Выплавка стали в электропечах”, “Непрерывная разливка стали” и “Производство агломерата”, которые используют только закупаемое сырье, так как им не предшествуют другие процессы рассматриваемого предприятия. К “Конечным переделам” относятся процессы “Прессование слитков” и “Холодный прокат листа”, так как за ними не следуют другие процессы этого предприятия. Выходы конечных переделов образуют выпускаемую для внешних потребителей продукцию. Но и некоторые из промежуточных переделов могут создавать конечную продукцию, например, “Сталепрокатный завод” при производстве проката наряду со слябами выпускает “Сортовой прокат”, предназначенный только для продажи, так как ни один из переделов его не использует. Упомянутый процесс “Производство проката” вместе с остальными (не упомянутыми) процессами относятся к промежуточным переделам, так как им предшествуют и за ними следуют другие процессы. Все это и многое другое видно из диаграммы на рис. 1.4.3. Например, анализируя эту диаграмму можно ответить на следующие вопросы:

- какую продукцию выпускает предприятие;
- что закупает предприятие;
- какие подразделения предприятия начинают производственный цикл;
- какие подразделения предприятия заканчивают производственный цикл;
- какие продукты являются внутренними;
- сколько переделов на предприятии;
- сколькими способами получают сталь в изложницах;
- как можно получить слябы;
- можно или нет временно приостановить работу электросталеплавильных цехов N2 и N3 без остановки производственного цикла.

## 1.5. Основные понятия финансовых подсистем

В производственном процессе продукты труда превращаются в товары. Роль специфического товара (*всеобщего товарного эквивалента*) выполняют *деньги*. В процессе реализации (продажи) товары превращаются в деньги. Основная функция финансово-кредитной подсистемы – так регулировать *финансовые потоки* (идущие в противоположную сторону по отношению к *товарным потокам*), чтобы обеспечить стабильный и справедливый обмен товарами и услугами как между экономическими объектами, так и между отдельными членами общества, а также обеспечить финансовые условия для развития производства. Деньги и другие ценные бумаги образуют финансовый ресурс экономики.

Теория финансов имеет дело с фирмами, рынками ценных бумаг и посредническими структурами. *Фирмы* обладают физическими (земля, машины и т. д.) и другими ценностями (организационные структуры, рынки и т. д.), они осуществляют производство и управление этими ценностями. *Посреднические структуры* – это коммерческие и инвестиционные банки, пенсионные фонды, страховые компании и т. д. Они выполняют финансовые поручения индивидуумов и фирм.

Любая ценность, подлежащая продаже, покупке или обмену, называется *активом*. *Финансовый рынок* определяется как совокупность таких активов. *Рынок ценных бумаг (финансовый рынок ценных бумаг)* – это совокупность организованных финансовых рынков для торговли, обмена и выполнения других операций с ценными бумагами. Принято выделять *основные ценные бумаги* – облигации (бонды) и акции – и *производные (вторичные) ценные бумаги* – опционы или контракты с опционами, варранты, фьючерсные контракты, ваучеры и др.

*Акции* – это долевыми ценные бумаги, выпускаемые фирмами с целью аккумуляции капитала. Цена акции определяется как текущим состоянием рынка ценных бумаг, так и деятельностью фирмы. Владелец акции, или акционер, получает право как на участие в управлении компанией (по правилу: число акций равно числу голосов), так и на получение дивидендов.

*Облигации* – это долговые обязательства, выпускаемые государством, банками, акционерными компаниями и другими финансовыми институтами с целью аккумуляции капитала. Примерами облигаций могут служить банковский счет, облигации государственного займа и т. д. В отличие от акций, облигации

выпускаются на некоторый срок, по истечении которого изымаются из обращения посредством погашения (выкупа). Характеристиками облигации являются время погашения, стоимость погашения (номинал), выплата до погашения (купоны). Обычно облигация – это безрисковый актив (по сравнению с акцией, цена которой достаточно хаотична), но опасность невыполнения долговых обязательств все-таки существует.

Для управления торговлей наличным товаром, различными контрактами и другими финансовыми инструментами, а также для предоставления информационных услуг создается бесприбыльная организация – *биржа*.

Акции и облигации являются первичными ценными бумагами, поскольку их цена определяется непосредственно через экономические факторы. В отличие от них вторичные (производные) ценные бумаги функционируют на базе уже имеющихся на бирже основных ценных бумаг. Рынок производных ценных бумаг является привлекательным из-за того, что требует существенно меньших начальных затрат. Одной из наиболее распространенных производных ценных бумаг является *опцион*, или *контракт с опционом*, – ценная бумага, дающая ее обладателю право продать (купить) некоторую ценность (например, акции, валюту и т.д.) на оговариваемых условиях. По времени исполнения (погашения) опционы делятся на два основных типа: европейские, имеющие фиксированную дату погашения, и американские, которые могут быть представлены к исполнению в любой момент до фиксированной даты.

Участника финансового рынка, помещающего свободные капиталы в те или иные активы, называют *инвестором*, а совокупность принадлежащих ему активов – *инвестиционным портфелем*. Искусство инвестора состоит в умении правильно и динамично формировать свой портфель инвестиций (управлять портфелем): хранить актив, покупать и продавать его, давать займы. Перераспределение портфеля служит для уменьшения риска той или иной сделки, например, покупки или продажи опциона. В этом случае говорят о хеджировании, т. е. защите своих инвестиций, а соответствующий динамический портфель называют хеджирующим портфелем.

Для обеспечения эффективности финансового процесса инвестор находит, например, такую инвести-

ционную стратегию, которая даст прибыль при нулевых начальных затратах. Такие стратегии называются *арбитражными*.

*Пример.* Пусть акция продается в Нью-Йорке по цене 200 долларов, а в Лондоне – по цене 100 фунтов. Текущий обменный курс доллара по отношению к фунту 1,71. В этом случае возможен следующий процесс:

- взять в долг 100 фунтов,
- купить акцию в Лондоне за 100 фунтов,
- продать ее в Нью-Йорке за 200 долларов,
- обменять 200 долларов по курсу 1,71 на 116,9 фунтов,
- вернуть долг 100 фунтов.

Результатом является чистая прибыль в 16,9 фунтов, и, следовательно, построена арбитражная стратегия.

Другой пример арбитража, имеющего российскую специфику, давала торговля *ваучерами* (приватизационными чеками).

Ясно, что арбитраж не может существовать длительное время, поскольку действия самих инвесторов, желающих им воспользоваться, приводят к исчезновению именно этой арбитражной возможности.

На рис.1.5.1 – 1.5.4 приведены диаграммы технических терминов, формально описывающие взаимосвязь понятий финансовых подсистем экономики. На рис.1.5.5, 1.5.6 приведены диаграммы, описывающие структуру рынков и инвесторов. На рис.1.5.7 изображена модель офисного процесса “Реализация арбитражной стратегии” на основе описанного сценария в рассмотренном примере. Здесь события изображены пиктограммой “взрыв”, функции изображены в виде стола с сидящим за ним исполнителем и т. д. Рядом с объектами приведены их названия (краткие имена). Этот тип диаграмм относится к процессному виду моделирования и является одной из разновидностей функционального потока, управляемого событиями. Здесь все объекты имеют визуально понятный смысл. Исключение составляют только логические операторы, т. к. они для каждого из трех допустимых значений (“И”, “ИЛИ” “ИСКЛЮЧАЮЩЕЕ ИЛИ”) изображаются в виде трех жирных стрелок, выстроенных в окружность. Для конкретизации логического оператора в ARIS можно выбрать другой тип их изображения. Но об этом в следующих главах.



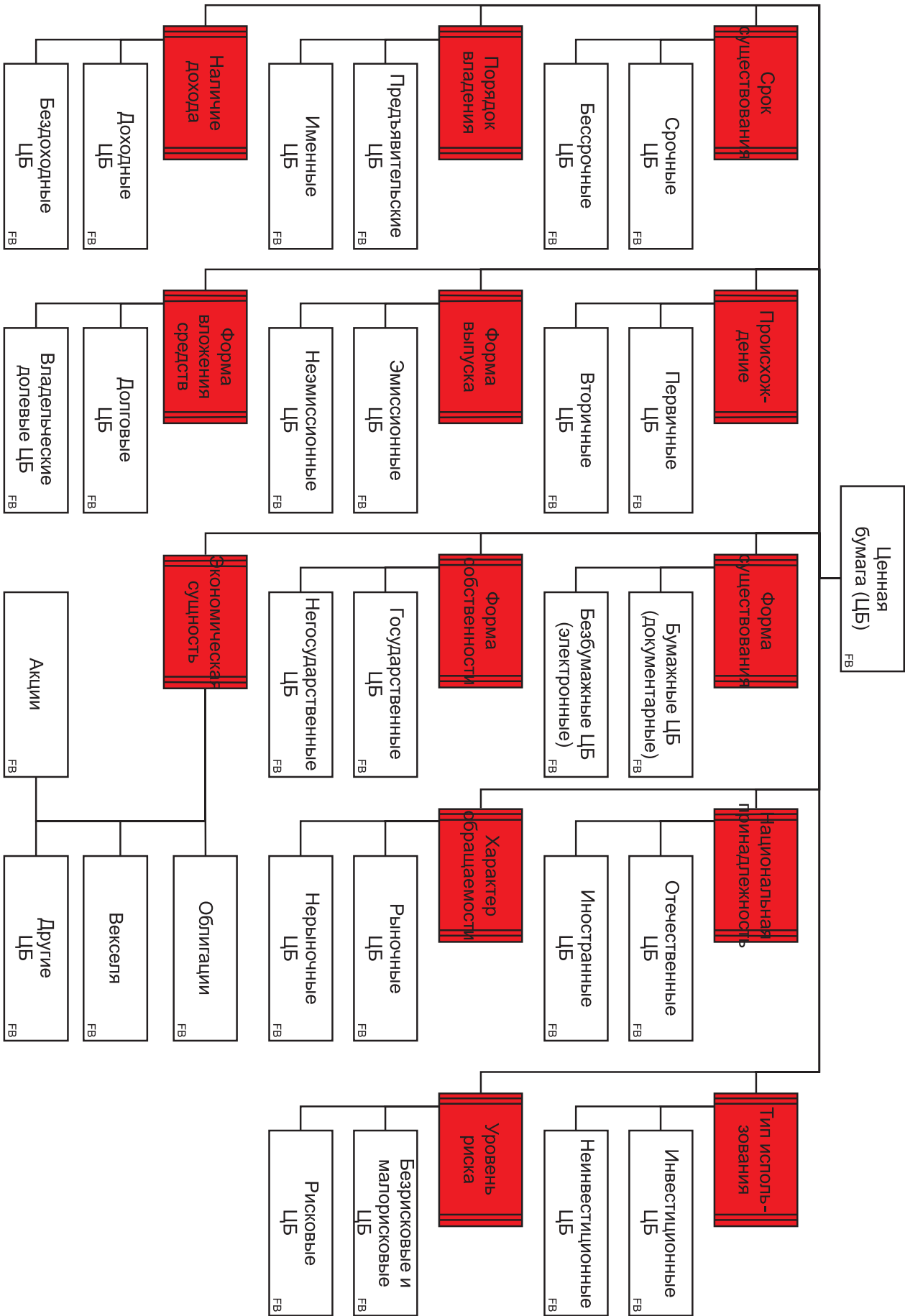


Рис. 1.5.1 Модель технических терминов, связанных с понятием “Ценная бумага”



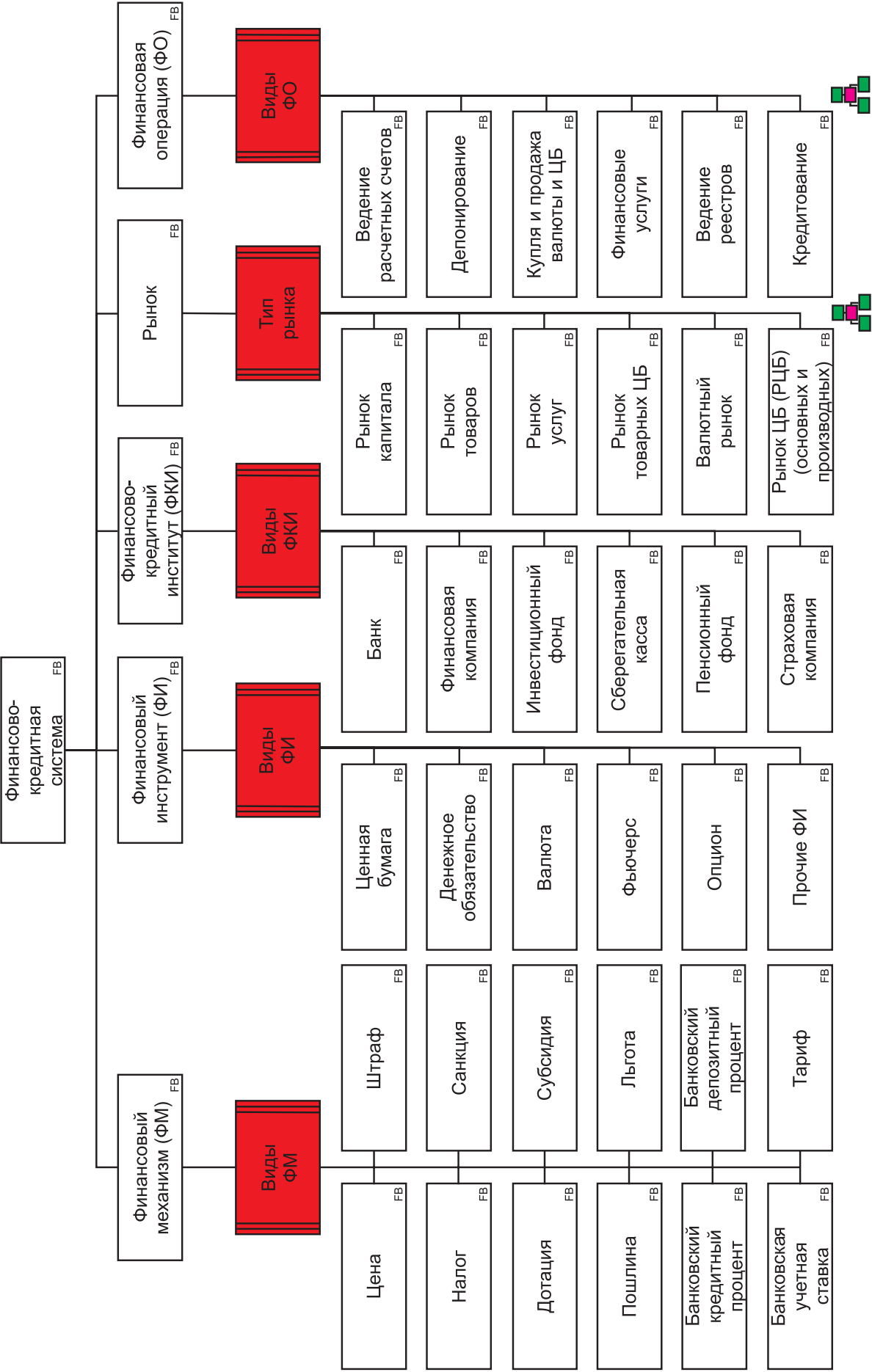


Рис. 1.5.2 Модель технических терминов понятия “Финансово-кредитная система” с детализацией понятий “Кредитование” и “Рынок ценных бумаг”

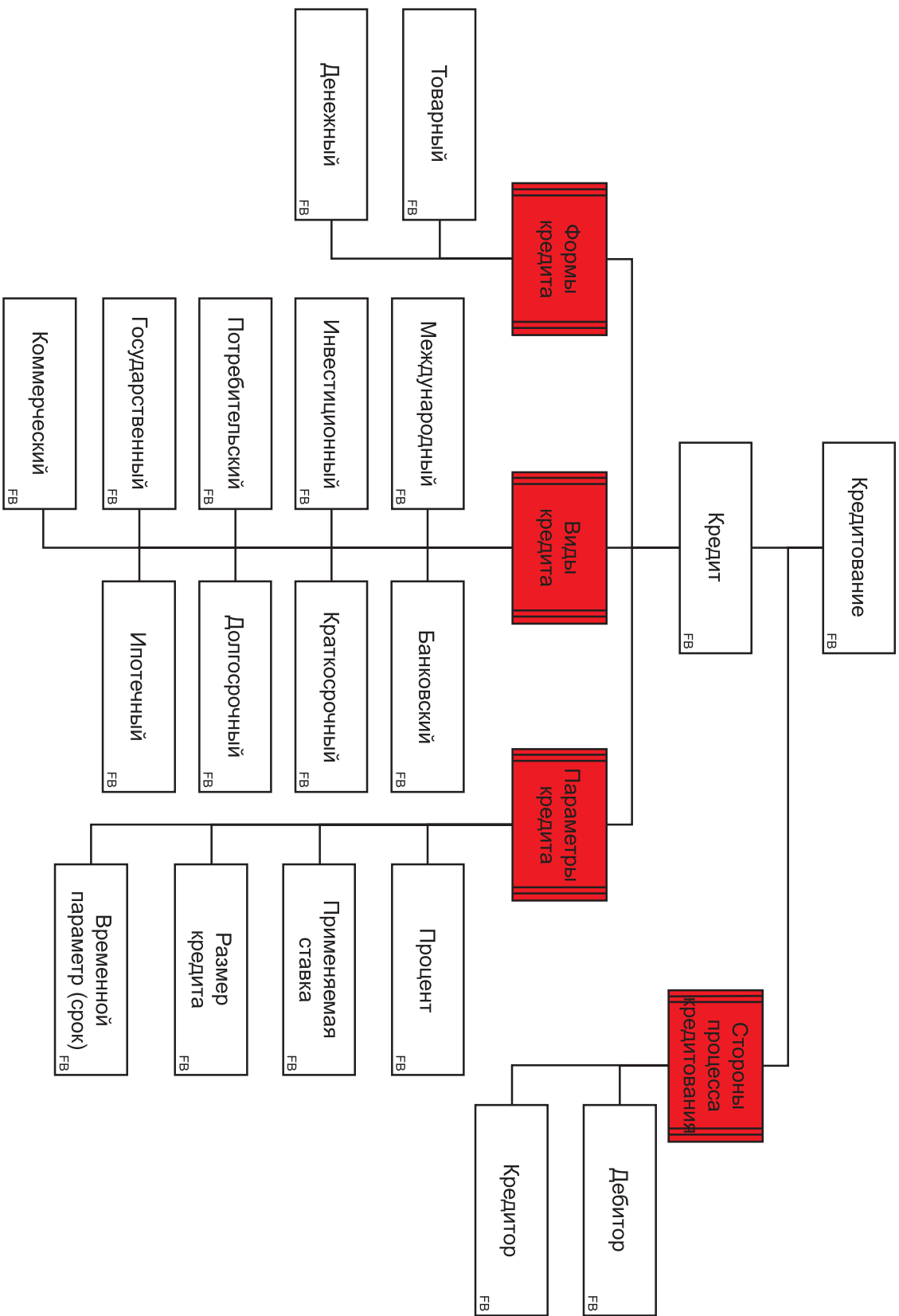


Рис. 1.5.3 Модель технических терминов понятия “Кредитование”

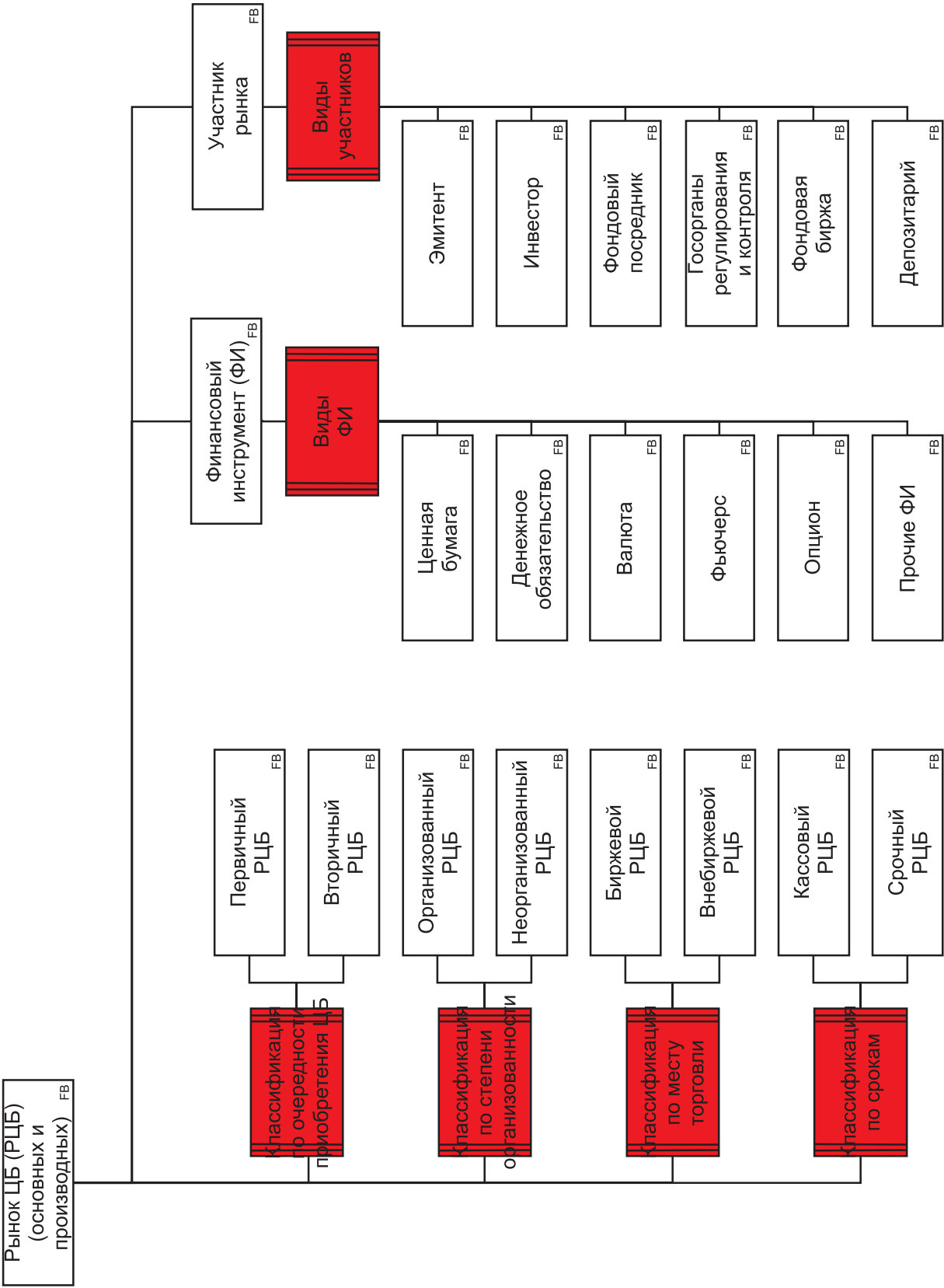


Рис.1.5.4 Модель технических терминов понятия “Рынок ценных бумаг (РЦБ)”

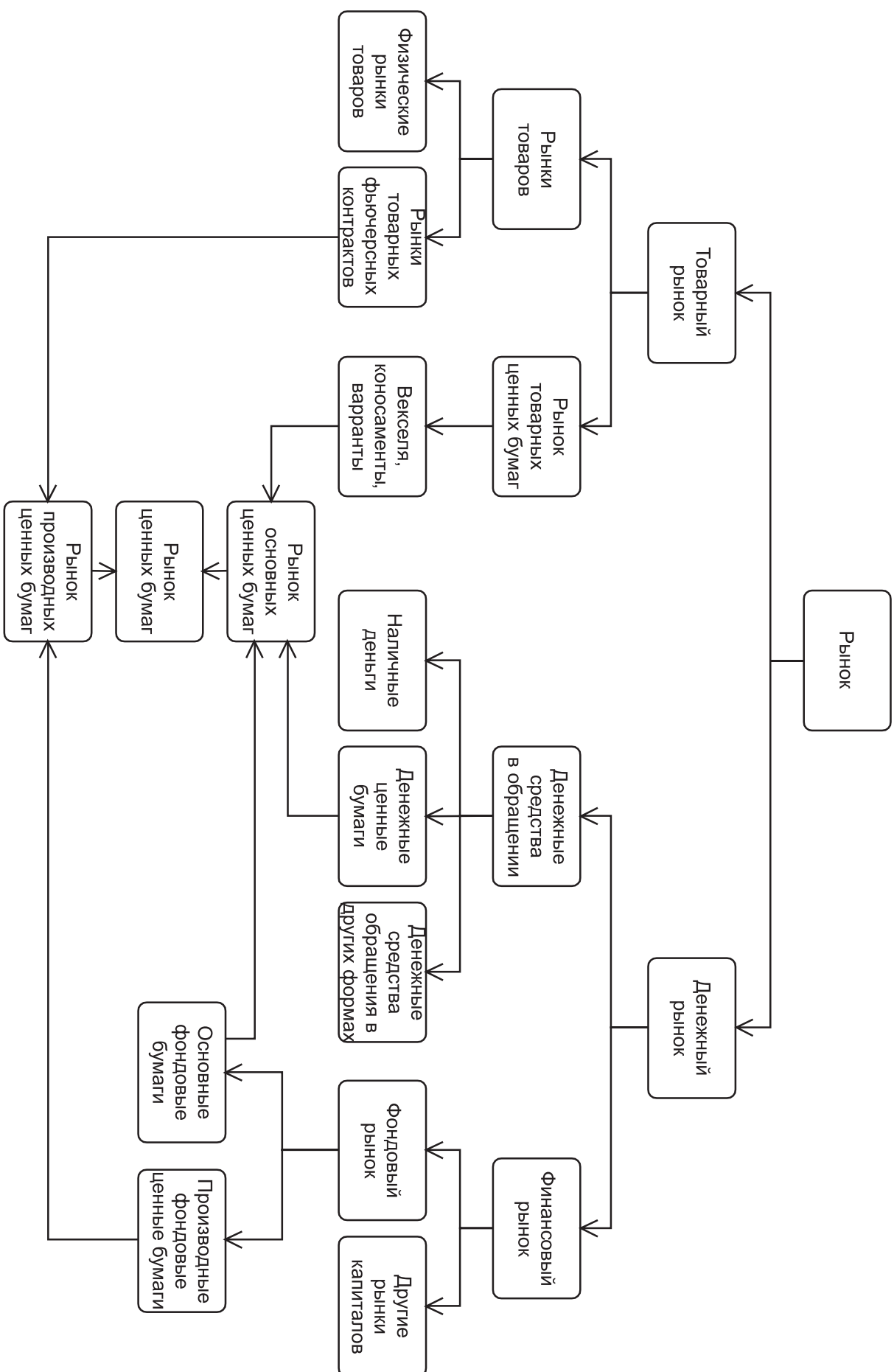


Рис.1.5.5 Структурная диаграмма рынков

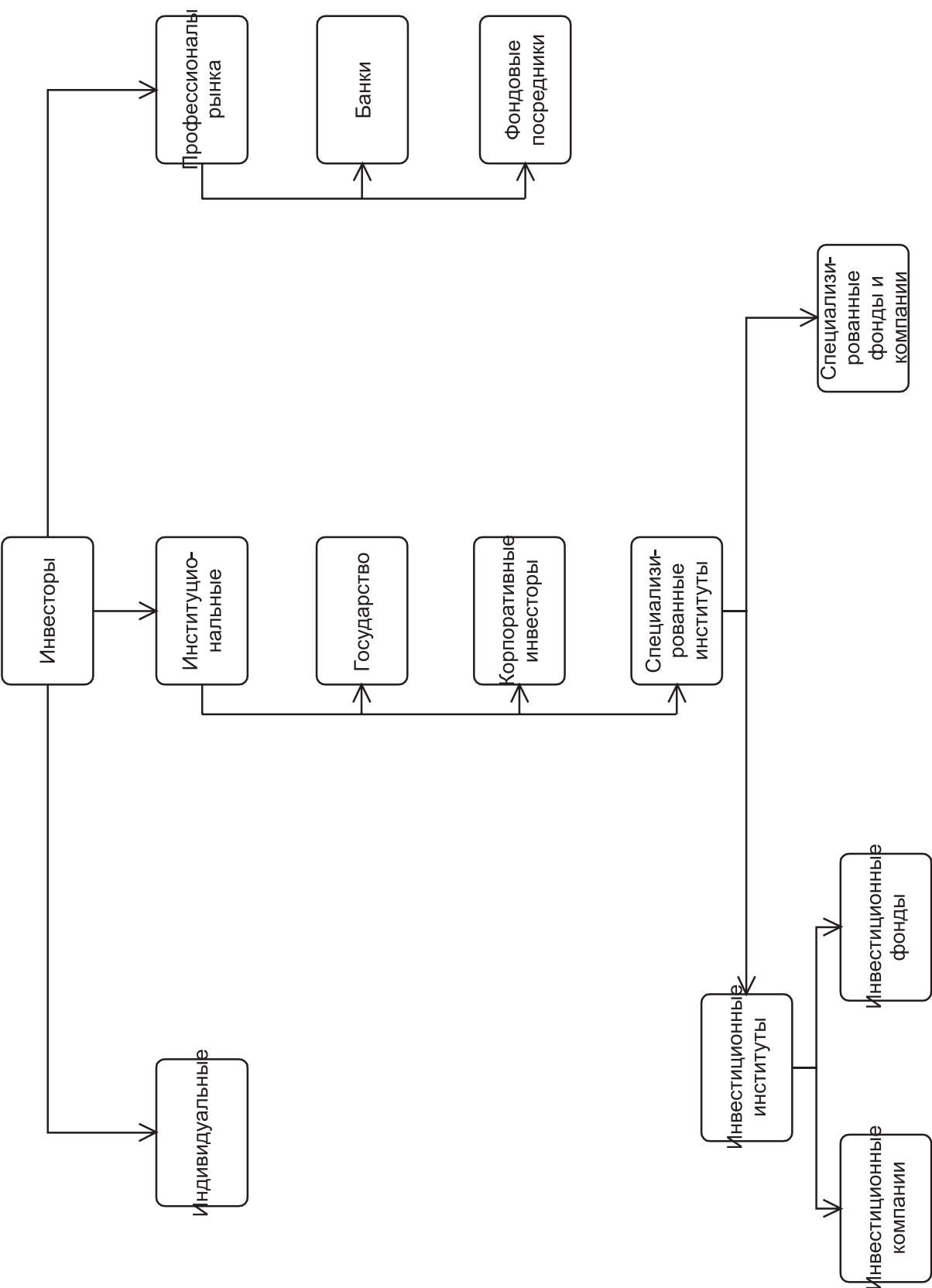


Рис. 1.5.6 Структурная диаграмма инвесторов

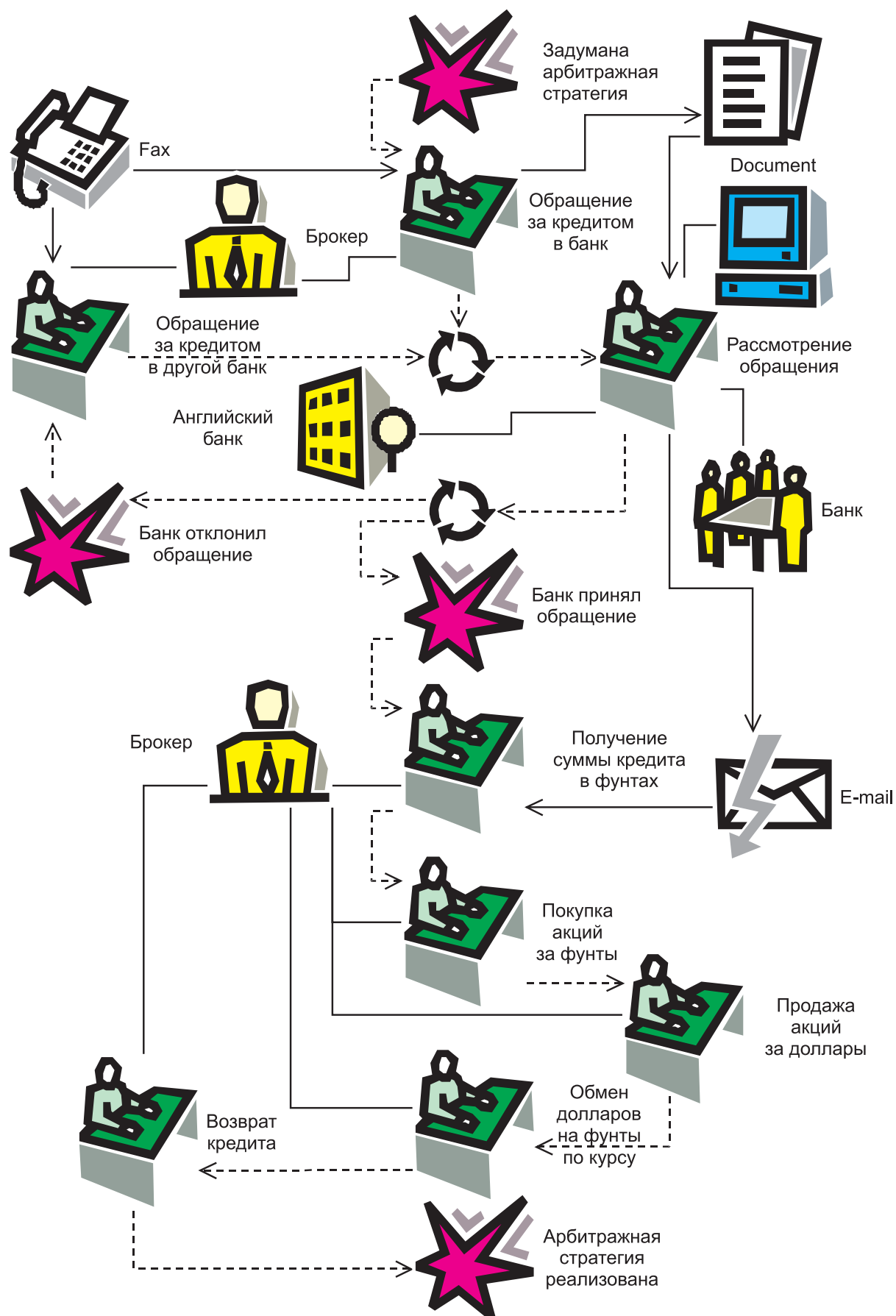


Рис.1.5.7 Офисный процесс “Реализация арбитражной стратегии”

## 2. ИНФОРМАТИКА В ЭКОНОМИКЕ

В этом разделе описываются основы информатики, раскрывается роль информационных систем в управлении предприятием. Кратко рассматриваются те понятия информатики, которые используются в экономике. Там, где возможно, вербальное изложение материала дополняется ARIS-диаграммами.

### 2.1. Некоторые понятия информатики

В обыденной жизни под термином “информация” понимают сведения, передаваемые людьми устным, письменным или другим образом.

*Информация* – совокупность фактов, явлений и событий, которые представляют интерес, регистрируются и обрабатываются. Это понятие объединяет источник (производителя) и приемник (потребителя) информации. Причем информация содержит факты, неизвестные ранее потребителю, и дополняющие его представление об изучаемой или анализируемой сущности (объекте, процессе, системе и др.). То есть информация – сведения, которые снимают существующую у потребителя неопределенность, расширяя понимание сущности полезными сведениями. По Шеннону, *информация* – это снятая неопределенность (Клод Шеннон – американский инженер и математик, родоначальник теории информации).

Формы представления информации различны. Основные из них – *символьная* (основанная на использовании символов – букв, цифр, знаков), *текстовая* (использует тексты – символы, расположенные в определенном порядке), *графическая* (различные виды изображений), *звуковая*.

Информация – это единственный неубывающий экономический ресурс, более того, ее объем постоянно возрастает. Особенно ярко это стало проявляться с середины 20-го века. В 70-е годы прошлого века объем информации удваивался каждые 5 – 7 лет. В 80-е годы удвоение происходило уже за 20 месяцев, а в 90-е – ежегодно. В начале 21-го века информация стала товаром первой необходимости, а расхожими стали, например, такие высказывания: “Кто владеет информацией, тот владеет миром” или “Он слишком много знал ...”.

Часто понятия “информация”, “данные” и “знания” рассматриваются как синонимы. Это неверно. *Данными* называется информация, представленная в удобном для обработки виде. *Знание* – это данные, полученные в результате познания действительности, достоверность которых проверена практикой.

В области систем искусственного интеллекта *знания* – это данные, на основании которых реализуется

процесс логического вывода. Другими словами, на основании этих данных можно делать различные заключения с помощью логического вывода. В системах обработки информации под знаниями понимают сложно-организованные данные, содержащие одновременно как *фактографическую* (регистрация некоторого факта), так и *семантическую* (смысловое описание зарегистрированного факта) информацию.

Приведем определения основных и часто используемых понятий информатики.

*База данных (БД)* – один или несколько специальным образом организованных файлов, хранящих систематизированную информацию, для доступа к которой используются программные средства *системы управления базой данных* (СУБД). БД представляет собой хранилище связанных между собой данных, не зависящих от программ, которые к ним обращаются. Например, в БД предприятия хранятся технологические, производственные, эксплуатационные, экономические и другие характеристики изделий предприятия и их аналогов, выпускаемых конкурирующими фирмами. Такой БД могут пользоваться специалисты из отдела маркетинга (для прогноза конкурентоспособности изделий на рынке), плановики (для определения необходимых объемов выпуска продукции), конструкторы (для анализа конструктивных особенностей изделий конкурентов) и т. д. БД должна быть достаточно полной, правильно организованной, актуальной и удобной для использования. Содержащиеся в ней данные не должны противоречить друг другу. Редактирование, пополнение и исключение данных, их поиск и сортировка осуществляются под управлением СУБД. БД могут быть персональными и коллективного пользования. Сопровождение крупных БД коллективного пользования осуществляет *администратор базы данных*.

*Администратор БД* – человек или группа лиц, ответственных за состояние, развитие и использование БД. Администратор БД обеспечивает работоспособность БД, контролирует и поддерживает полноту, правильность, непротиворечивость и *целостность данных*, необходимый уровень *защиты данных*. Он взаимодействует с пользователями и программистами, программы которых используют доступ к БД.

*Система управления базой данных (СУБД)* – комп-

лекс программ и языковых средств, предназначенных для создания, ведения и использования БД. СУБД дает программистам возможность разрабатывать новые, более совершенные средства *обработки данных*, а пользователям – возможность непосредственно управлять данными в БД. К наиболее известным СУБД для персональных компьютеров относятся системы FoxPro фирмы Fox Software и Access фирмы Microsoft. Важнейшей характеристикой СУБД является *модель данных*, положенная в основу БД. В настоящее время наибольшее распространение получили реляционные СУБД. Упомянутые СУБД являются реляционными.

*Модель данных* – совокупность принципов *организации БД*. Самыми популярными являются иерархическая, сетевая и реляционная модели данных. Каждая модель данных предусматривает различные принципы определения, манипулирования и хранения данных в БД, но наиболее важным является принцип организации связей между данными в БД. В иерархической модели данных записи данных связаны между собой явным образом, причем способ связывания строго ограничен, а именно, записи данных в иерархической БД образуют *иерархию*. Например, *реестр Windows* является иерархической БД. В сетевой БД записи также явным образом связаны друг с другом, но способ связей не ограничен. В реляционной модели данных записи связаны друг с другом неявно, через значения *атрибутов (полей кортежа)*.

*Иерархическая база данных* – БД, организованная по иерархической *модели данных*. Иерархическая БД имеет более простую структуру и эффективную реализацию по сравнению с другими БД.

*Реляционная база данных* – БД, организованная по реляционной *модели данных*. Она состоит из таблиц, каждая из которых представляет собой множество *кортежей* одинаковой структуры. Таковую таблицу можно интерпретировать как отношение (в математическом смысле) между *атрибутами* таблицы. Связи между кортежами различных таблиц в реляционной БД устанавливаются неявно (по совпадению значений атрибутов в разных таблицах).

*Сетевая база данных* – БД, организованная по сетевой *модели данных*. Она состоит из наборов записей, которые связаны между собой так, что некоторые записи могут содержать явные ссылки на другие наборы записей. Тем самым, наборы записей образуют сеть. Сетевая БД отличается от *иерархической БД* тем, что связи между записями могут быть совершенно произвольными, и отличается от *реляционной БД* тем, что связи явно присутствуют и хранятся в БД.

*Распределенная база данных* – БД, в которой данные распределены между несколькими узлами *компьютерной сети*. Не следует путать распределенную БД с *сетевой БД*.

*Банк данных* – система файлов и БД, предоставляющая услуги по хранению и поиску данных в одной предметной области (например, статистика торговли). Пользователи банка данных могут находиться в разных учреждениях и даже регионах, получая доступ к данным по *компьютерной связи* или по почте.

*База знаний (БЗ)* – один или несколько специальным образом организованных файлов, хранящих систематизированную совокупность понятий, правил и фактов, относящихся к некоторой предметной области. Содержимое БЗ оформляется, связывается между собой и представляется таким образом, чтобы на его основе можно было с помощью специальных программ осуществлять рассуждения и делать выводы, получая сведения, которые в явном виде могут не присутствовать в БЗ. Для построения БЗ применяются методы *искусственного интеллекта*, специальные языки описания знаний и *интеллектуальный интерфейс*. БЗ являются основной содержательной частью *интеллектуальных информационных систем, интеллектуальных обучающих систем, интеллектуальных систем программирования и экспертных систем*, где с их помощью представляются навыки и опыт экспертов – специалистов в данной предметной области.

*Информационная система (ИС)* – любая *система*, связанная с накоплением, хранением или *обработкой информации*. В этом смысле ИС является и систематизированная картотека, и *банк данных*. Под ИС понимается также *вычислительная система*, предназначенная для хранения, поиска и выдачи информации по запросам пользователей (людей и программ). Обычно ИС включает в себя большие и сложные БД и БЗ и обеспечивает информацией пользователей из нескольких организаций. Существуют информационно-поисковые системы, в которых поиск и отбор информации осуществляется по заданным в запросе признакам или условиям, и информационно-справочные системы, работающие в *интерактивном режиме* и обеспечивающие пользователей сведениями справочного характера.

*Информационная среда* – хранящаяся в компьютере, но не оформленная в виде ИС совокупность знаний, фактов и сведений, относящаяся к некоторой предметной области и используемая одним или несколькими пользователями.

*Информационный продукт* – совокупность дан-



ных, которые сформированы производителем для распространения (продажи, обмена) в вещественной или невещественной форме.

*Информационный ресурс* – организованная совокупность документированной информации, включающая БД, БЗ и другие массивы информации в ИС (архивах, фондах и пр.).

*Информационный рынок* – система экономических, правовых и организационных отношений по торговле продуктами интеллектуального труда на коммерческой основе. В отличие от торговли обычными товарами, имеющими материально-вещественную форму, на рынке информационных продуктов в качестве предмета продажи или обмена выступают ИС, информационные технологии, лицензии, патенты, товарные знаки, ноу-хау и прочие виды информационных ресурсов экономики.

*Информационные технологии* – это совокупность методов, способов, приемов и средств, которые ориентированы на получение, обработку и распространение (передачу) информации. Информационные технологии проникли во все сферы человеческой деятельности. Они в своем развитии прошли несколько этапов, которые достаточно условно можно назвать так: ручной (перо и чернильница), механический (пишущая машинка), электрический (электрическая машинка, ксерокс), электронный или компьютерный. В технологии обработки информации первичные сведения (о производственных операциях, исполнителях, выпускаемой продукции, фактах приобретения и продажи товаров) являются предметом труда, а получаемая результатная информация (используемая для анализа и принятия управленческих решений) является продуктом труда. В качестве средств труда здесь, как правило, выступают ПЭВМ и программные системы, автоматизирующие обработку информации.

*Информационные процессы* – процессы сбора, передачи, накопления, хранения, обработки, поиска, выдачи и доведения информации до пользователя.

*Пользователь (потребитель) информации* – субъект, обращающийся к ИС или посреднику за получением необходимой информации.

На рис.2.1.1 приведена ARIS-диаграмма технических терминов, формально описывающая взаимосвязь некоторых из введенных понятий.

Наступивший век “подарит” нам *информационное общество*, т. е. общество, в котором большинство работающих будет занято *производством, преобразованием, передачей, обработкой, хранением, тиражированием, распространением, накоплением и реализа-*

*цией информации*, особенно высшей ее формы – знаний.

## 2.2. Экономическая информация в управлении

Для управления экономической системой (подсистемой) используется, как правило, человеко-машинный комплекс со следующими основными подсистемами обеспечения (рис.2.2.1).

1. *Информационное обеспечение* – экономическая ИС, методы классификации и кодирования информации, технология обработки данных, нормативно-справочная информация, система документооборота и др.

2. *Организационное обеспечение* – описание процессов функционирования системы управления, инструкции обслуживающему персоналу, должностные инструкции и др.

3. *Техническое обеспечение* – комплекс технических средств, включающий компьютеры и средства связи.

4. *Математическое обеспечение* – совокупность методов, правил, математических моделей и алгоритмов решения управленческих задач.

5. *Лингвистическое обеспечение* – совокупность терминов, искусственных языков, правил формализации естественного языка и др.

6. *Программное обеспечение* – совокупность программных систем обработки данных, документов и решения управленческих задач.

7. *Правовое обеспечение* – совокупность правовых норм, определяющих создание, юридический статус и функционирование системы.

Остановимся подробнее на информационном обеспечении (см. диаграмму на рис.2.2.2).

Информация, используемая для управления, называется *управленческой*. Это разнообразные сведения экономического, технологического, социального, демографического и другого содержания. В экономических процессах управленческая деятельность является основной, а информация выступает как один из важнейших ресурсов наряду с природными, производственными, трудовыми и финансовыми.

Важнейшей составляющей управленческой информации является *экономическая информация*, т. е. совокупность сведений, используемых при управлении экономической системой (подсистемой) и обеспечивающих производство, распределение, обмен и потребление товаров и услуг. Словосочетание *экономическая информация* вошло в обиход в 60-х годах 20-го столетия с внедрением ЭВМ в сферу управления экономической. Экономическую информацию классифицируют по

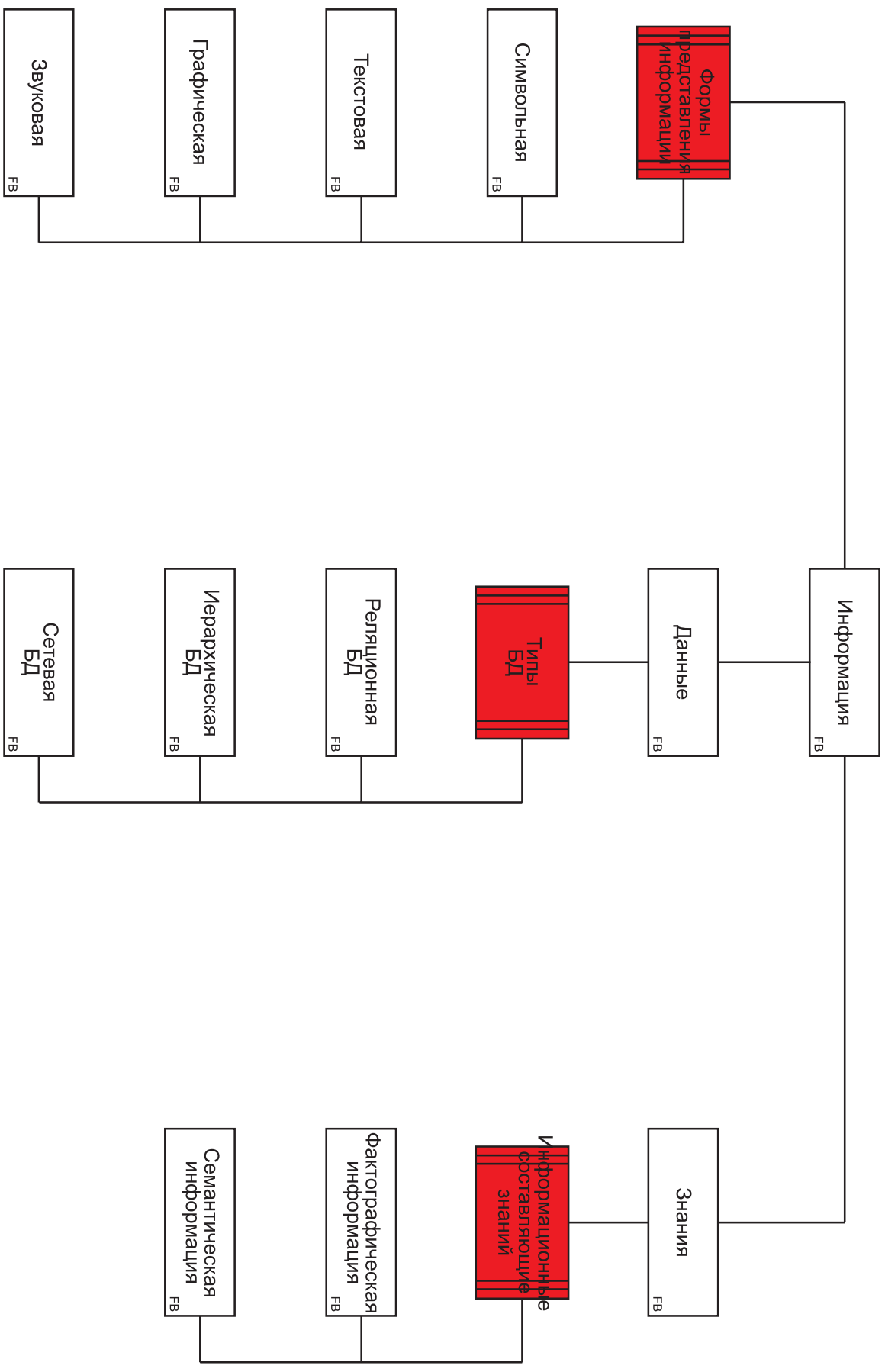


Рис.2.1.1 Модель терминов, связанных с понятием “Информация”

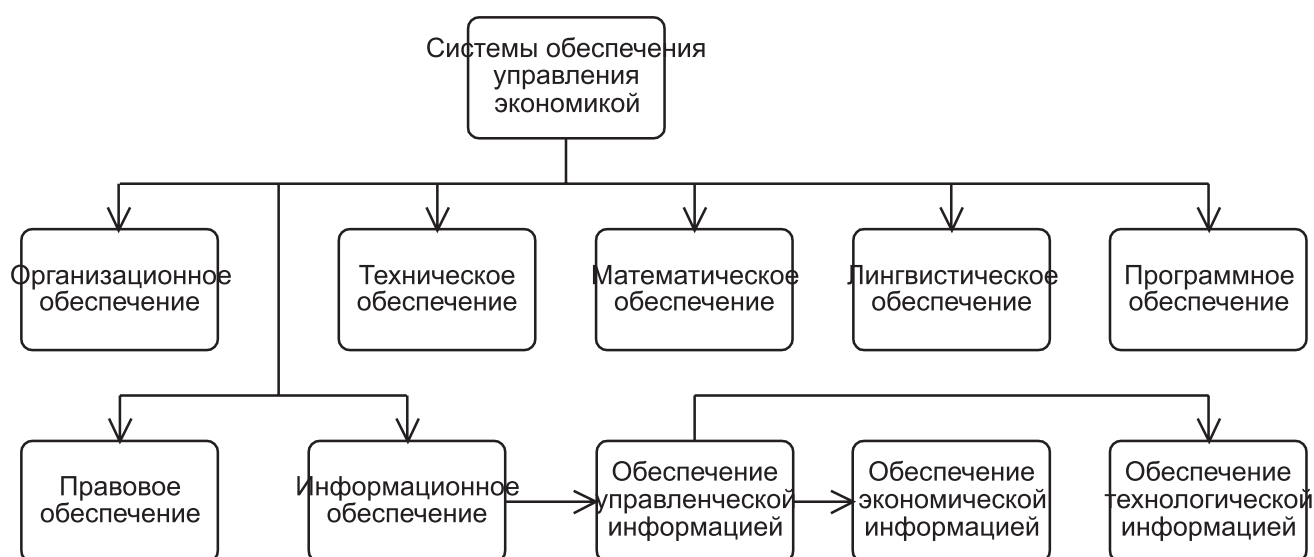


Рис.2.2.1 Структурная диаграмма систем обеспечения управления экономикой

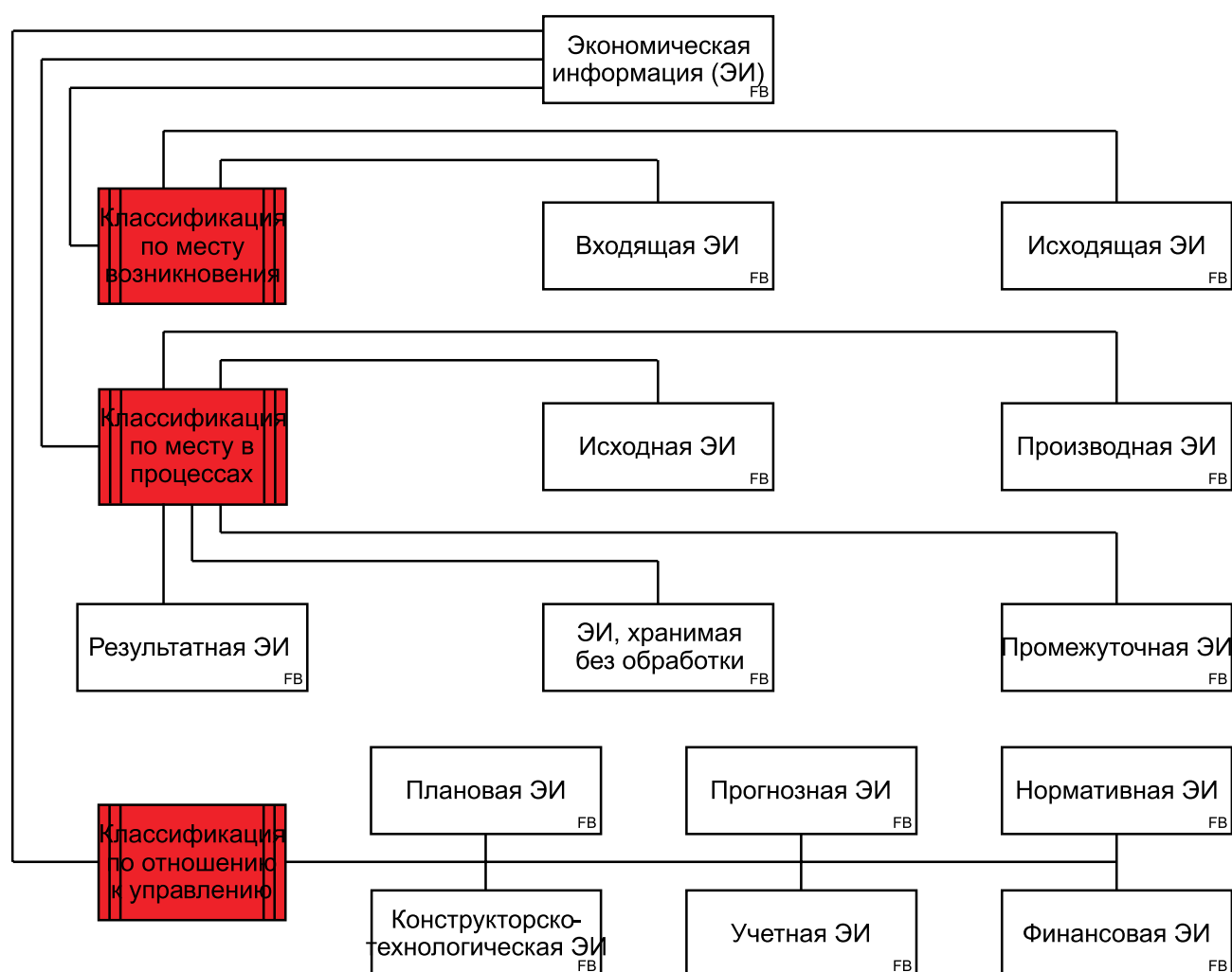


Рис.2.2.2 Модель классификации экономической информации

месту возникновения (входящая, исходящая), по участию в процессе обработки/хранения (исходная, производная, хранимая без обработки, промежуточная, результатная), по отношению к функциям управления (плановая, прогнозная, нормативная, конструкторско-технологическая, учетная, финансовая и др.) и пр.

Экономическая информация является составляющей ИС предприятия, которая прежде всего должна быть ориентирована на внутренние нужды руководства самого предприятия. Информация для инвесторов, налоговых органов, кредиторов и т. д. тоже важная составляющая экономической ИС. Квалифицированная подготовка данных для внешней среды требует как минимум наличия и доступности самих данных. А это как раз и обеспечивает экономическая ИС.

Какой бы сложной и “разумной” не была ИС, ее применение бесполезно, если входные (первичные) данные неточны. Роль и значение первичной информации переоценить невозможно. Поэтому экономист должен владеть технологией работы с первичной информацией.

Для регистрации хозяйственной операции, т. е. для получения первичной информации об экономическом процессе, необходимо выполнить такие действия, как идентификация, привязка ко времени, измерение.

**Идентификация** – это действие (процесс), в результате которого устанавливают (узнают, определяют) идентификатор сущности. В качестве последней может выступать и субъект труда (человек, выполняющий операцию), и объект труда (обрабатываемая деталь), и объект передачи (что передано), и субъект передачи (от кого, кому), и пр. **Идентификатором** называется комбинация признаков, поставленная в соответствии с рассматриваемой сущностью и однозначно отличающая ее от любой другой сущности в данной ИС. Другими словами, идентификатор – это уникальное имя сущности. Идентификатором может быть цифровой код получателя, признаки защиты денежного знака и многое другое. В зависимости от конкретных обстоятельств идентифицировать требуется, например, либо только вид объекта (например, модель холодильника, достоинство банкноты, сорт ткани), либо и вид, и экземпляр объекта (работник предприятия с его уникальным табельным номером, смаркард).

**Привязка ко времени (датирование)** – процесс, в результате которого фиксируется время/дата (возможно, начало и завершение) выполнения операции.

**Измерение** – это нахождение какой-нибудь меры значения какой-либо величины. Способы, средства и единицы (штуки, килограммы, литры, рубли) измерения зависят от измеряемой сущности. В процессе изме-

рения формируются первичные данные.

Процесс получения первичных данных имеет ряд особенностей. Прежде всего следует учитывать, что сбор данных есть обычный *трудовой процесс*, и он требует определенной квалификации, а также затрат сил и времени. Первичные данные должны точно описывать первичные хозяйственные операции, т. е. первичная информация должна быть достоверна. Она должна быть еще своевременной и ценной.

**Ценность (или полезность)** информации проявляется в том случае, если она способствует достижению стоящей перед потребителем цели. Ценность информации – свойство относительное: одна и та же информация имеет разную ценность для разных потребителей. С течением времени ценность информации уменьшается – она стареет. Но старит информацию не время, а появление новой информации, которая отвергает полностью или частично имеющуюся информацию, уточняет ее, дополняет, дает новое сочетание сведений, приводящее к получению дополнительного эффекта.

Экономическая информация в документах, используемых для принятия управленческого решения или являющаяся результатом такого решения, как правило, структурирована. Рассмотрим некоторые структурные компоненты экономической информации.

Экономические показатели описывают разные сущности (предмет, процесс, явление, объект). Каждая сущность имеет определенные свойства (вес, габариты, цена и пр.). Совокупность сведений, отражающих какую-либо сущность, называют *информационной совокупностью*. Обычно информационная совокупность имеет иерархическую структуру. Например, “Данные о поставщике” включают его “Ф.И.О.”, “Адрес”, “Номенклатуру товаров”, “Условия поставки”. “Адрес” содержит “Почтовый индекс”, “Город” и т. д.

Степень детализации информационной совокупности конечна. Не делимая далее на смысловые единицы информационная совокупность получила название *реквизит* (часто используются синонимы: *слово, элемент данных, атрибут*).

**Реквизиты (документов)** – совокупность формальных элементов документа, отсутствие которых лишает документ юридической силы. Хотя реквизит является основным элементом экономической информации (дата, сумма, наименование и т. д.), он, взятый отдельно, не имеет экономического смысла. Различают два вида реквизитов: *реквизит-признак* и *реквизит-основание*. Если реквизит описывает качественное свойство информации (время или место действия, Ф.И.О. исполнителя и т. д.), то его называют *реквизит-признак*.

Если же реквизит представляет количественную характеристику (объем продукции в штуках, цена в рублях и пр.), то его называют реквизит-основание.

Сочетание одного реквизита-основания с одним или несколькими соответствующими ему реквизитами-признаками образует показатель. *Показатель* – качественно определенная величина, дающая количественную характеристику отображаемой сущности (предмету, процессу, явлению, объекту) и имеющая экономический смысл. Это информационная совокупность наименьшего состава, достаточная для образования самостоятельного сообщения или формирования документа. Например, информационная совокупность “пять пар женской обуви” состоит из реквизита-основания “пять” и трех реквизитов-признаков: “пара”, “женская” и “обувь”, которые имеют экономический смысл и поэтому являются показателями. Совокупность логически связанных реквизитов, имеющая юридическую силу, называется *документом* (см. рис.2.2.3).

### 2.3. Прикладная информатика в экономике

Что такое информатика? Чем информатика отличается от других наук о построении вычислительных систем? Что отличает информатику от кибернетики? Почему информатика в экономике выделяется в отдельную дисциплину? Постараемся ответить на эти вопросы.

*Информатика* – это наука об описании, осмыслении, интерпретации, представлении, формализации и применении информации (данных, знаний) с помощью ЭВМ для поиска новой информации (новых данных, знаний) во всех областях деятельности человека.

Термин информатика введен во Франции в 60-х годах прошлого столетия путем слияния двух слов: *information* (информация) и *automatique* (автоматика) для обозначения области, занимающейся автоматизированной обработкой информации. Понятия “кибернетика” и “информатика” различны. Основная цель кибернетики связана с разработкой теории управления сложными динамическими системами. Информатика же занимается изучением процессов получения и преобразования информации для решения задач управления (см. рис.2.3.1).

В кибернетике *управление (control)* – процесс целенаправленного воздействия на объект, обеспечивающий желаемое функционирование объекта, например, движение по заданной программе. В экономике *управление (management)* – процесс принятия управленческого решения и контроля его реализации с целью обеспечения желаемого функционирования экономической

системы (подсистемы). Например, при управлении процессом развития стремятся выполнить программу развития предприятия. Разработкой экономических программ (планов) занимаются специалисты – работники плановых отделов предприятий.

Кибернетика – это наука об управлении, точнее, о системах управления. Кибернетическую систему (т. е. систему, исследуемую кибернетикой) отличает от “некибернетической” наличие в ее составе объекта управления и системы управления. Система управления может определенным образом воздействовать на объект управления для достижения поставленных целей. Кибернетика предоставляет нам теорию и правила проектирования систем управления. Системы управления ракетой или нервной системой в живом организме являются кибернетическими системами.

Опыт применения компьютеров в управлении сколько-нибудь сложными экономическими процессами показал, что управлять ими на основе принципов кибернетики невозможно. Процесс управления коллективами людей выходит за рамки кибернетики – ее аппарат в этой ситуации “не работает”. Так как на место формализуемой реакции на изменение внешних условий ставится принятие управленческого решения, при котором неизбежно неформальное согласование различных интересов и критериев “оптимальности” множества людей, участвующих в процессе управления в условиях заведомо неполной информации. Этот неформальный элемент и необходимость согласования интересов (в том числе ущемления чьих-то интересов!) подразумевают *персональную, личную ответственность* за принимаемые решения. Поэтому нет экономической кибернетики, а есть менеджмент. Поэтому в управлении экономической упор делается не на *управляющие*, а на *информационные* системы, которые предназначены не реагировать на изменение ситуации непосредственно, а своевременно предоставлять человеку (лицу, принимающему решение) информацию, достаточную для адекватной реакции на происходящие события.

Информатика и “наука о компьютерах” – не одно и то же. Проектирование ИС в экономике подразумевает знание не только возможностей входящих в их состав аппаратных и программных средств, но и прежде всего основ экономики, особенно такого ее раздела, как теория менеджмента. Роль экономических закономерностей в построении ИС переоценить невозможно.

Специфика управления коллективами людей не позволяет применить к ним аппарат кибернетики в сколько-нибудь нетривиальных случаях. Поэтому для управления людьми типична не управляющая, а

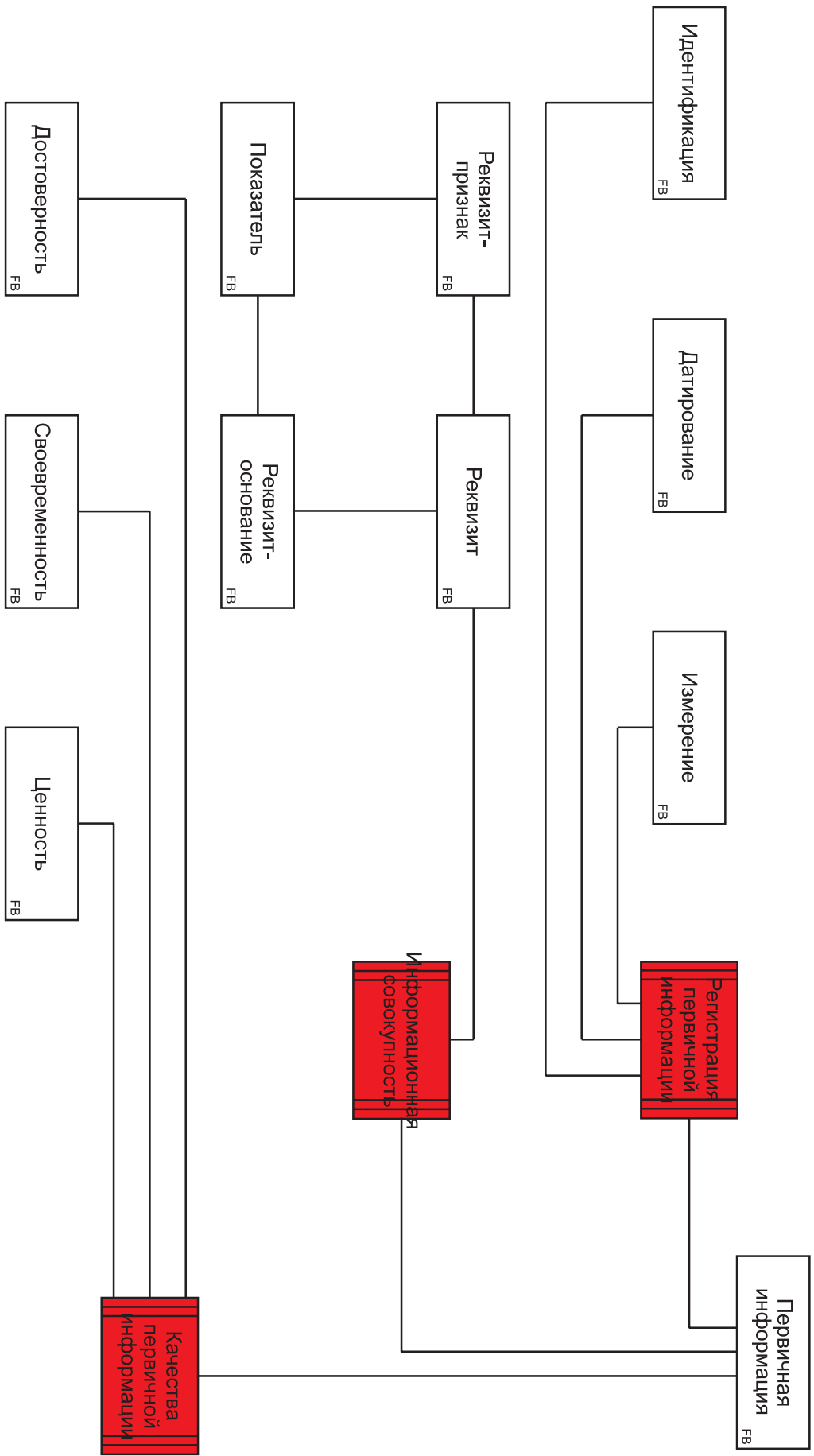


Рис.2.2.3 Модель терминов, связанных с понятием “Первичная информация”



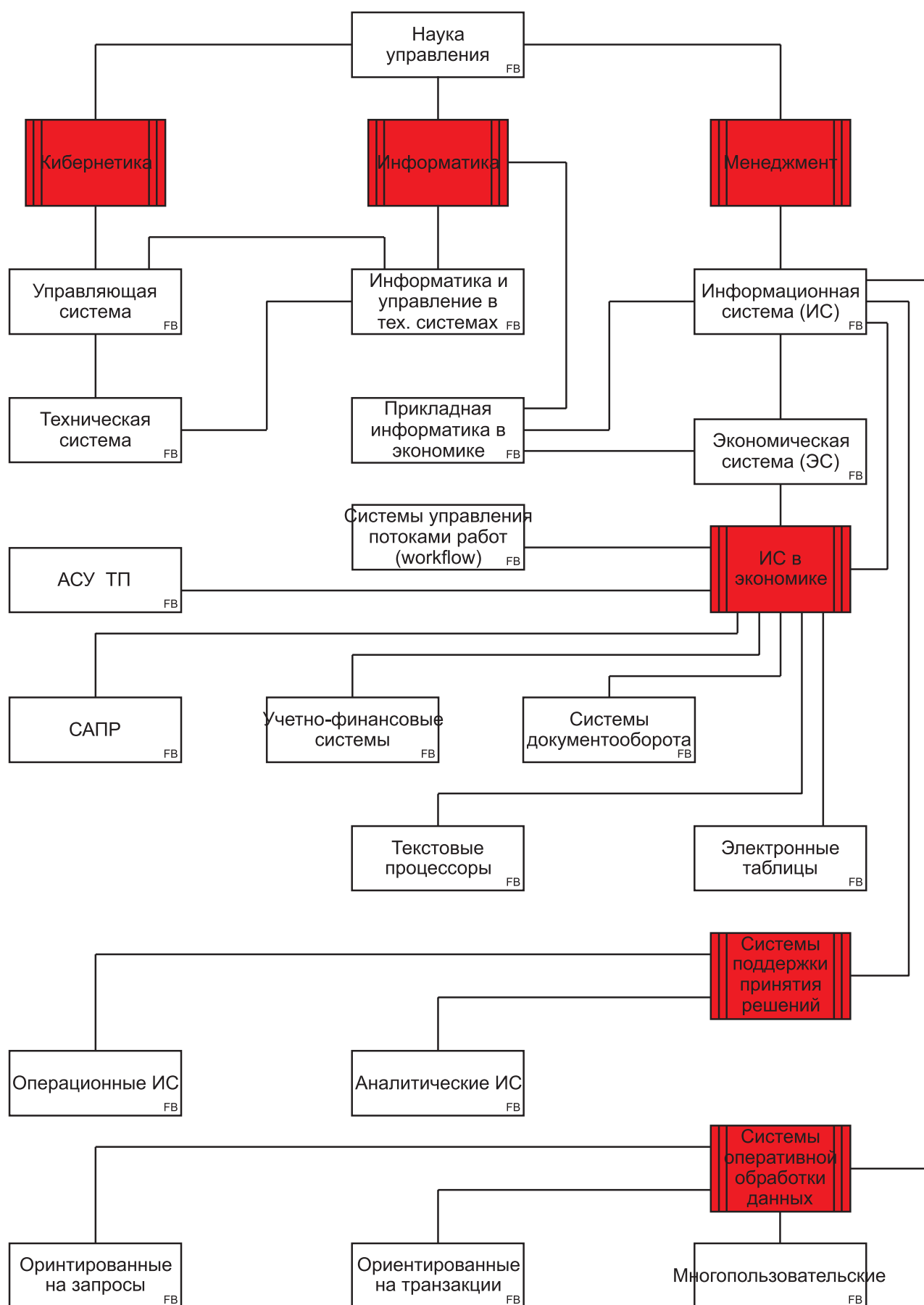


Рис. 2.3.1 Модель технических терминов, связанных с науками об управлении

информационная автоматизированная система, теоретические же основы таких систем изучаются не кибернетикой, а информатикой. ИС в экономике являются предметом специальности “Прикладная информатика в экономике”.

Перед ИС стоят совершенно иные задачи, чем перед системами управляющими, поэтому и законы их построения различны. Вместе с тем ИС имеет столь же строгие законы построения, как, например, дом или автомобиль. С другой стороны, ИС призвана упростить и облегчить взаимодействие людей в процессе управления. Поэтому различных ИС со своими закономерностями будет столько, сколько существует разных по своей сути процессов управления. Именно по этой причине информатика имеет множество отраслевых подразделов, значительно больше, чем кибернетика. В то же время ИС имеют значительное количество общих черт, что и позволяет считать их все предметом одной общей дисциплины – информатики.

Персональные компьютеры и появившиеся вместе с ними программы общего назначения, прежде всего текстовые редакторы и электронные таблицы, создали феномен персональной ИС – информационной среды, выстраиваемой пользователем любой квалификации, исходя из собственных потребностей и уровня подготовки. Это поставило перед информатикой задачу разработки подходов к интеграции этих персональных ИС в единую корпоративную (и глобальную) информационную среду, важное место в которой занимают единые и централизованно хранимые БД.

Остановимся на одной из проблем, возникающей при отсутствии единой БД предприятия. Например, пусть на предприятии имеются бухгалтерская система и система финансового анализа со своими БД. Поскольку финансовая система не может не пользоваться бухгалтерскими данными, они должны быть введены вручную как в бухгалтерскую, так и в финансовую системы. Помимо избыточной потребности в дисковом пространстве, возникает гораздо более серьезная проблема целостности данных – данные в программе финансового анализа могут отличаться от таковых в бухгалтерской программе, например, из-за ошибок оператора, неодновременности обновления данных и т. д. Контроль таких несоответствий очень труден, и в долгосрочной перспективе проще и дешевле создать единую ИС, объединяющую бухгалтерскую и финансовую службы, а в конечном счете – единую ИС масштаба предприятия.

Открывшаяся возможность автоматизировать деловое общение в целом как процесс потребовала от информатики дальнейшей специализации в соответ-

ствии со специализацией этих процессов. Например, управление производством (автоматизированное управление технологическими процессами), автоматизированное проектирование, прикладные программы общего назначения (текстовые редакторы, электронные таблицы и т. д.), системы автоматизации деловых процедур и документооборота, учетно-финансовые системы. Эти большие группы ИС различаются по таким критериям, как структура применяемых данных, количество пользователей, одновременно имеющих доступ к данным, степень ориентированности на транзакции и запросы (*транзакция* – обновление данных в системе, *запрос* – выборка данных из системы по заданным пользователем приказам и в заданной пользователем форме). Здесь речь идет о критериях, определяющих глубинные, сущностные характеристики ИС, поэтому законы их построения и эксплуатации также будут различны. Например, современный менеджмент выдвигает свои требования к построению ИС.

Заметим, что творческие, неформальные задачи, как правило, сконцентрированы на верхних этажах иерархии управления, тогда как для нижних этажей типично выполнение некоторых формальных процедур в соответствии с инструкциями. Большой объем первичных данных предъявляет повышенные требования к агрегированию данных. Участие в процессе множества людей и возможность случайных или намеренных посягательств на данные заставляет уделять значительное внимание проблеме безопасности данных.

ИС подразделяются на операционные и аналитические. Операционные ИС ориентированы на накопление и передачу больших массивов данных по стандартным процедурам и методикам. Аналитические ИС строятся во многом по иным правилам. В самом деле, если операционные ИС ориентированы, в первую очередь, на ввод новых данных, то аналитические – на их извлечение. Как следствие, число пользователей аналитических ИС на несколько порядков меньше числа пользователей операционных ИС, так как извлечение данных требует значительно более высоких полномочий, чем их ввод по стандартной процедуре. Если операционная ИС ориентирована во многом на ограничение полномочий пользователей по соображениям безопасности, то аналитическая – на максимально возможное расширение доступных пользователям средств, особенно в части формирования сложных запросов к ИС и т. д. Эти и многие другие аспекты построения и эксплуатации ИС в экономике являются предметом специальности “Прикладная информатика в экономике”.

## 2.4. Конфиденциальная информация и ее защита

Информация, охранительный правовой режим которой устанавливается ее собственником на основе законов, называется *конфиденциальной информацией*. Для обеспечения безопасности бизнеса весь объем деловой информации делят на открытую (доступную всем) и конфиденциальную, т.е. предназначенную для ограниченного (“своего”) круга лиц.

Особо значимая часть конфиденциальной информации составляет коммерческую тайну. *Коммерческая тайна* – это сведения, которые не являются государственными секретами, но разглашение которых может нанести ущерб бизнесу. Эти сведения связаны, например, с производством, технологической или конструкторской документацией, новыми разработками, финансами.

Действующим законодательством предусмотрено, что определенные сведения не могут быть коммерческой тайной (учредительные документы, информация о платежеспособности, сведения о наличии вакантных рабочих мест и т. д.).

На рис.2.4.1 изображена диаграмма технических терминов для понятий рассматриваемой предметной области. В первых трех ярусах этой диаграммы представлены введенные уже понятия. Дадим некоторые пояснения к диаграмме на рис.2.4.1.

К кластеру “Объекты конфиденциальных интересов” относятся те объекты, которые являются предметом противоправных интересов отдельных людей или организаций. К следующему кластеру отнесены семь источников, которые владеют или содержат в себе конфиденциальную информацию. В кластере “Цели посягательств на информацию” выделены три цели преступного посягательства на информацию. В последнем кластере “Способы несанкционированного доступа” перечислены четырнадцать основных способов несанкционированного доступа к конфиденциальной информации. Предпоследний кластер составляют способы защиты информации, где все начинается с создания необходимой правовой основы. Для каждого из трех элементов этого кластера на рис.2.4.2 – 2.4.4 представлены детализирующие их модели. В организации должны быть разработаны и введены в действие нормативные документы, которые регламентируют меры защиты в соответствии с действующим законодательством. Без создания правовых основ обеспечения безопасности любые последующие претензии предпринимателя к кому-либо могут оказаться просто беспочвенными.

С учетом значимости ЭВМ необходима специаль-

ная комплексная защита средств вычислительной техники. Строится эта защита по тем же принципам, что и защита информации в системе целиком: организационные меры защиты, аппаратные, программные и криптографические средства защиты.

Следует заметить, что охота за информацией и защита информации – это вечная борьба копия и щита, которая издавна идет с переменным успехом и которую еще долго придется вести.

Определения для понятий в моделях можно получить в ARIS-отчете (смотрите раздел 7.5). Он автоматически формируется по специальной программе (скрипту) системой ARIS 5.0 на основе диаграммы модели и ее атрибутов “*Определение объекта*”, в которых эти определения занесены разработчиками диаграмм.

В отчетах по представленным здесь моделям технических терминов изложено, что должны регламентировать *нормативные правовые документы*, какие *организационные мероприятия* направлены на затруднение несанкционированного доступа к конфиденциальной информации, какие *средства инженерно-технической защиты информации* применяются на практике и многое другое. Ниже приведены некоторые определения, которые содержатся в атрибутах *Description/Definition* каждого из технических терминов моделей на рис.2.4.1-2.4.4.

*Люди.* К этой категории относятся рабочие, обслуживающий персонал, продавцы продукции, клиенты, партнеры, поставщики, посредники, покупатели и т. д. Люди не только обладают и распространяют информацию, они способны ее анализировать, обобщать, делать выводы, а также скрывать или совершать какие-либо криминальные действия с информацией. Одним из лучших источников конфиденциальной информации являются “болтуны”. Поэтому на съездах специалистов, конференциях, научных семинарах, семинарах по обмену опытом умельцы собирают самую свежую и ценную информацию.

*Документы.* Самая распространенная форма накопления, хранения и обмена информацией. Они весьма разнообразны по своему функциональному назначению, содержанию и материальным носителям. Обычно о сохранности и защите документов проявляют особую заботу.

*Публикации.* Информационные носители в виде открытых изданий: книги, статьи, доклады, рекламные проспекты и т. д. По оценкам специалистов, до 60 % секретной военной информации и до 95 % важной промышленной информации можно получить из открытых источников.

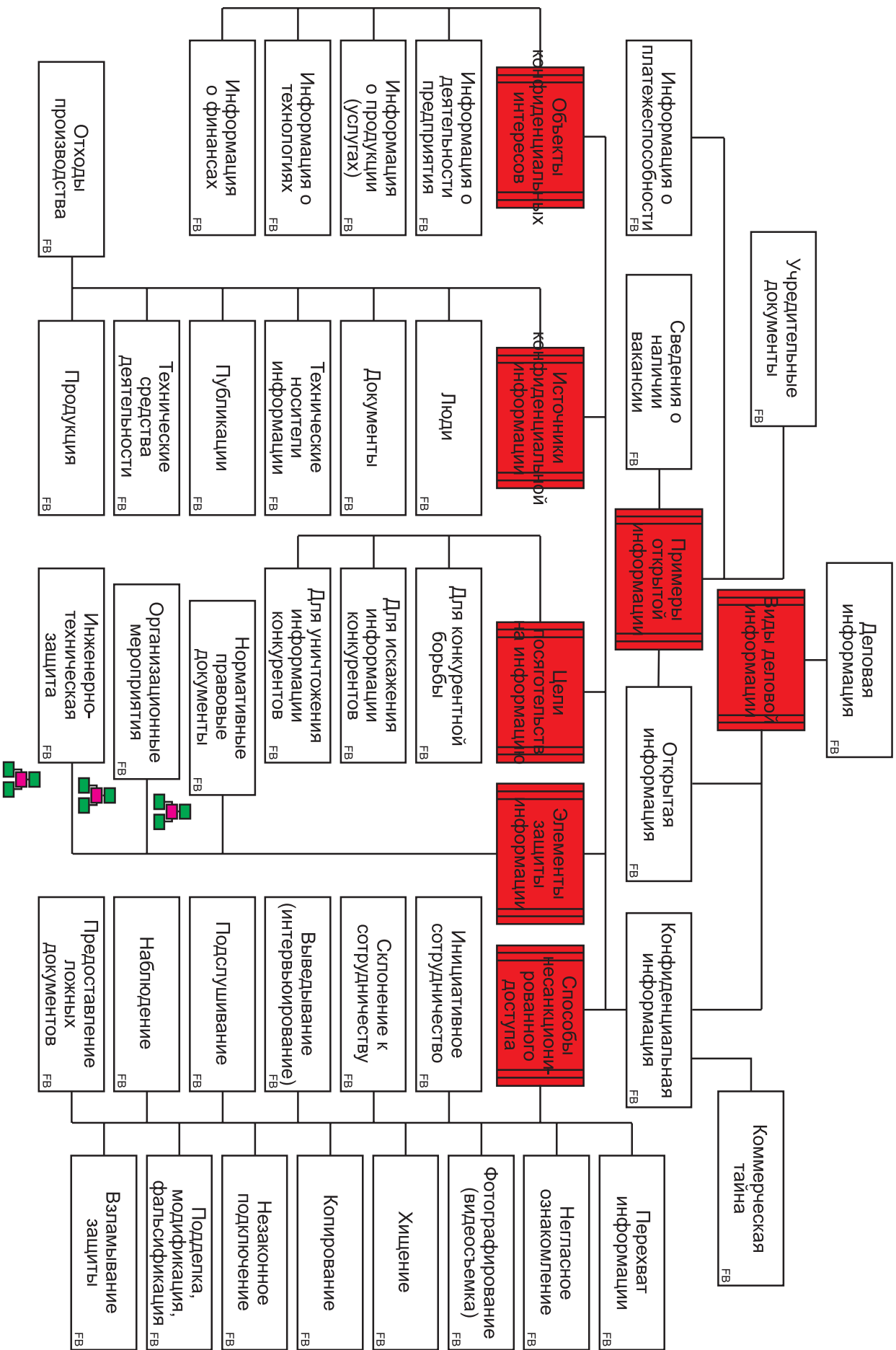


Рис.2.4.1 Модель технических терминов понятия “Деловая информация”

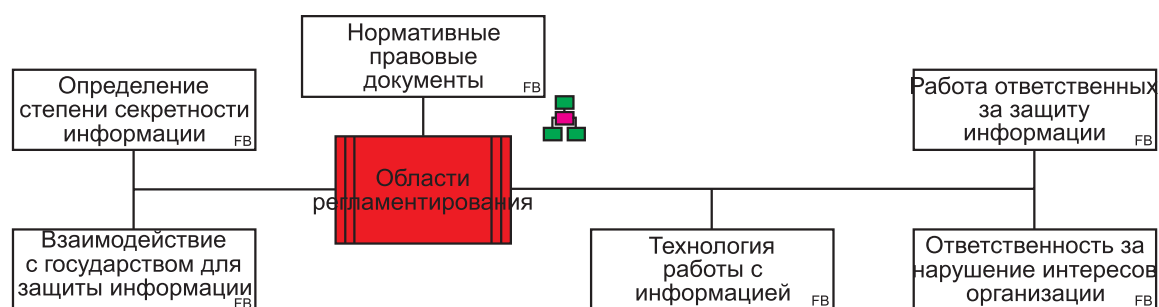


Рис.2.4.2 Модель технических терминов понятия “Нормативные правовые документы”

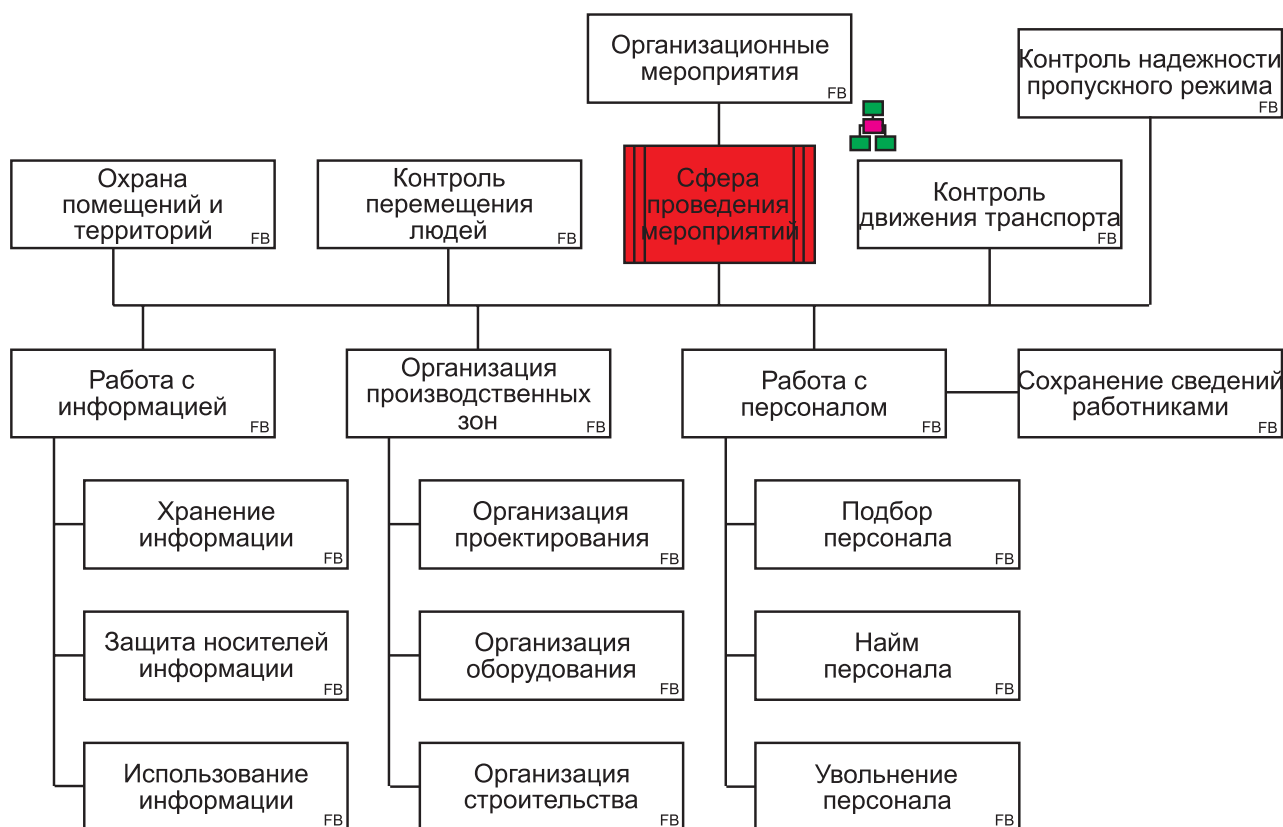


Рис. 2.4.3 Модель терминов “Организационные мероприятия для защиты информации”

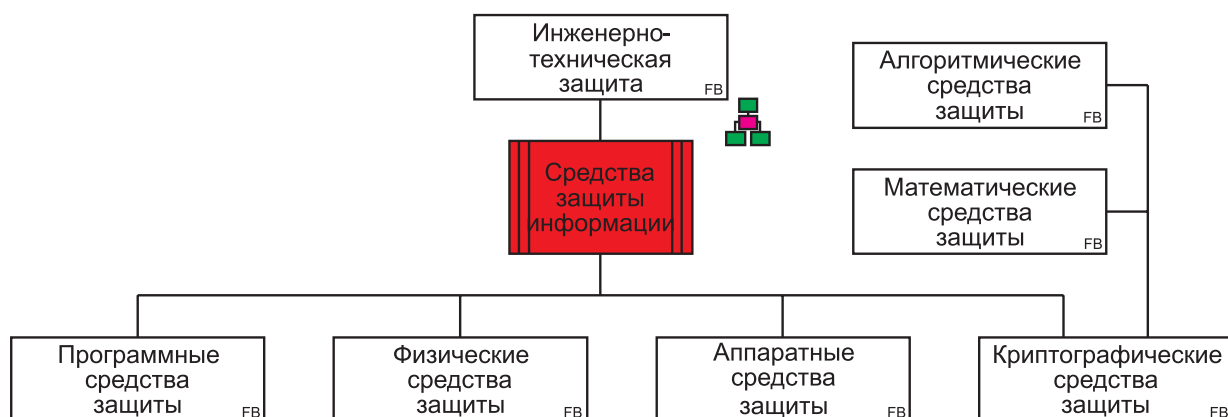


Рис. 2.4.4 Модель терминов понятия “Инженерно-техническая защита”

*Технические носители информации.* Бумажные носители, кино- и фотоматериалы, магнитные носители, аудио- и видео-носители, распечатки программ и данных на принтерах и экранах ЭВМ, табло индивидуального или коллективного пользования и др. Опасность утечки информации, связанная с техническими носителями, в том, что их парк (как и парк ЭВМ) растет очень быстро, широко доступен и трудно контролируем.

*Технические средства обеспечения производственной деятельности.* В эту группу входят средства и каналы связи, охранные и пожарные системы, трансляционные системы, автоматизированные системы сбора и обработки информации и ряд других систем. Полнота и достоверность информации источников этой группы делают их весьма привлекательными для злоумышленников.

*Продукция.* Весьма специфический источник информации. Каждый этап жизненного цикла продукции (замысел, разработка, опытный образец, испытания и доводка, серийное производство, эксплуатация, модернизация, снятие с производства) сопровождается своей информацией, проявляющейся в виде разных физических эффектов, демаскирующие признаки которых могут раскрыть охраняемые сведения.

*Промышленные и производственные отходы.* Отходы могут быть очень ценным источником информации, тем более что они могут зачастую добываться легально, безопасно и легко. Выброшенные копирки, черновики и т. д. могут рассказать много интересного специалисту, а химический анализ отходов может открыть очень важный секрет.

*Инициативное сотрудничество.* Часто сотрудник организации сам идет на противоправные действия. По мнению специалистов, около 25 % работников готовы в любое время и при любых обстоятельствах предать интересы организации, в которой работают, и около 50 % готовы сделать это в зависимости от обстоятельств. Эти обстоятельства весьма разнообразны, иногда даже неожиданны: финансовые затруднения; политическое, религиозное, научное, моральное инакомыслие; недовольство зарплатой, статусом, должностью; обиды на начальство или власть; тщеславие, амбиции, зависть, непомерные запросы или семейные проблемы; вредные пристрастия и многое другое. Опасность инициативного сотрудничества трудно переоценить, так как такое сотрудничество особенно тяжело раскрывается, а проникает такой сотрудник обычно очень глубоко внутрь организации.

*Склонение к сотрудничеству.* Это, как правило,

насилованное действие со стороны злоумышленника по отношению к работнику данной организации, осуществляемое путем подкупа, запугивания, провокации, шантажа, устройства ловушек и т. д. Одной из разновидностей склонения к сотрудничеству иногда бывает переманивание специалистов конкурента к себе для овладения его знаниями. Незаконность, бесчестность, подлость и омерзительность средств и способов склонения к сотрудничеству не знает границ.

*Выведывание (интервьюирование).* Стремление с помощью внешне наивных, невинных вопросов получить определенные сведения. Речь идет о милом, ненавязчивом, но очень эффективном разговоре и скрытом выуживании информации. Осуществляется выпытывание разным образом: официальные прессконференции и интервью, беседы в ближайшем окружении интересующего человека, ложное трудоустройство и ложный прием на работу и т. д. Выпытывание – достаточно эффективный способ доступа к конфиденциальной информации, и есть большие умельцы его использования.

*Подслушивание.* Способ ведения разведки и шпионажа, применяемый как агентами, так и специальными постами подслушивания. Подслушивание – весьма распространенный способ получения информации, способ многогранный, со многими приемами и средствами. К достоинствам этого способа добывания информации надо отнести то, что получаемая информация имеет определенную эмоциональную нагрузку (она иногда бывает не менее важной, чем прямое содержание) и поступает в реальном масштабе времени, без опозданий и задержек (если идет непосредственное подслушивание). Технические способы реализации подслушивания чрезвычайно разнообразны, оригинальны, сложны и достаточно дороги.

*Наблюдение.* Способ ведения разведки визуально или с помощью технических средств. Наблюдение – многогранный и многоплановый способ ведения разведки. Квалифицированное наблюдение требует привлечения хороших специалистов и технических средств. Зато результаты наблюдений, в случае успеха, могут быть весьма впечатляющими и доказательными.

*Хищение.* Умышленное противоправное овладение информацией. Это широко используемый злоумышленниками способ посягательства на чужое добро. Как считают специалисты, 10 % людей никогда не воруют в силу своих моральных воззрений, 10 % воруют при каждом удобном случае при любых обстоятельствах, 80 % людей, как правило, честны, но могут поддаться соблазну украсть. Воруют, как правило, “плохо лежащую” информацию. Надо иметь в виду, что не



всегда хищение информации обнаруживается быстро и не всегда похищенное можно восстановить, поэтому хищение – достаточно опасное нарушение прав законных владельцев информации.

**Копирование.** Как способ получения конфиденциальной информации чрезвычайно распространено и популярно, так как оставляет законного владельца в неведении о случившемся. Существуют многочисленные эффективные технические способы и приемы копирования как документов, так и машинной памяти. Как и при хищении, копируют все то ценное или потенциально ценное, что плохо лежит.

**Подделка, модификация, фальсификация.** Способы получения доступа к конфиденциальной информации путем предоставления ложных документов; изменений, которые ведут к преодолению запретов; искажению истинной картины в ущерб законным владельцам. При этом могут возникнуть ситуации, когда права первооткрывателей и законных владельцев защитить очень трудно.

**Уничтожение информации.** Осуществляется с применением технических средств или программным путем. Для уничтожения информации иногда бывает достаточно поразить системы автоматизированной обработки данных, защиты информации, связи, охраны, противопожарной защиты и др.

**Незаконное подключение.** Контактное или бесконтактное получение тех сообщений, которые передаются по различным линиям и проводам. Подключиться для перехвата каких-то сообщений можно практически к любым линиям связи, даже к сетям питания и заземления, даже к волоконно-оптическим линиям связи. Обнаружить подключение очень сложно.

**Перехват.** Получение информации за счет электромагнитных волн пассивными средствами приема. Перехватываются не только сигналы радиосвязи, но и другие электромагнитные излучения (излучения радиолокационных и радионавигационных систем, систем телеуправления, сигналы ПЭВМ, возникающие при выдаче на экран электронно-лучевой трубки текущей информации и др.). Возможен перехват переговоров по радиотелефону и по беспроводным системам связи внутри учреждения. Возможная дальность установления контакта колеблется от десятков метров до тысяч километров (в зависимости от характера источника и используемых средств перехвата). Поскольку растет объем информации, передаваемый “радиоспособом”, постольку растет и значение такого способа незаконного получения информации, как перехват.

**Негласное ознакомление.** Это любой способ полу-

чения информации, к которой субъект не допущен. Одним из обстоятельств, способствующих негласному ознакомлению с конфиденциальной информацией, является низкая трудовая дисциплина и пренебрежение правилами режима секретности. Для негласного ознакомления сейчас широко используются многочисленные весьма совершенные технические средства. Часто используют феноменальные возможности человека (зрительная память, слух, умение читать речь по губам и т. д.).

**Фотографирование (видеосъемка).** Способ получения видимого изображения на разнообразных фотоматериалах. Особенность способа – документальность. Снимки могут иметь разное назначение, от компромата до раскрытия секретов технологического характера.

**Физические средства защиты.** Препятствуют проникновению злоумышленников на объекты и защищают персонал, материальные ценности, финансы и информацию от противоправных действий.

**Аппаратные средства защиты.** Приборы, устройства, приспособления и т. д., которые призваны препятствовать утечке информации.

**Программные средства защиты.** Специальные программы, программные комплексы и системы защиты информации в информационных системах различного назначения и средствах обработки данных.

**Криптографические средства защиты.** Специальные математические и алгоритмические средства защиты информации, передаваемой по сетям связи или хранимой и обрабатываемой на ЭВМ, которые основаны на шифровании информации.

## 2.5. Офис как информационное предприятие

Основной функцией работы служащих в офисе является предоставление нужной информации нужному человеку в нужное время. Работа служащих в офисе складывается из чтения документов, их подготовки, размышления над изложенными в документах фактами и общения с клиентами. Возникающая в офисе информация всегда имеет адресата – клиента, использующего эту информацию.

Понятие офиса имеет материальный (конторское помещение и оборудование) и организационный (формы и структура управления) компоненты. Офис может быть самостоятельным учреждением, входить в более крупную организационную структуру, либо быть информационно-управляющей ячейкой в сфере производства или обслуживания. Важной особенностью работы офиса является то, что это не только источник конеч-

ных информационных услуг, но и источник решений, регламентирующих поведение людей или распределение материальных ресурсов. Существует несколько определений понятия “офис”. Приведем два из них.

Офис – это организационная единица, на входе и выходе которой не материальные изделия, а “бумаги” (письма, заявления, распоряжения, контракты, соглашения, заказы на покупку, счета на оплату, запросы на оказание услуги и др.).

Офис – это служба типа канцелярии, функция которой получать и выпускать “документы”, а также отслеживать их движение, где под “документами” понимают любое сообщение, передаваемое друг другу участниками процесса.

В общем случае офис вырабатывает не только “бумаги” и траекторию их перемещения, а решения, имеющие ценность для клиента. Поэтому офис следует рассматривать как полноценное информационное предприятие (часто пользующееся правом юридического лица), преобразующее информационные ресурсы в информационные продукты.

*Электронный офис* воплощает концепцию всестороннего использования в деятельности компьютеров и средств связи. Электронный офис практически исключает внутрифирменные представления документов на бумаге и концентрирует деятельность вокруг специалиста или руководителя.

Электронный офис, благодаря электронной почте и компьютерам, увеличивает возможности взаимодействия людей без их физической концентрации в пределах одного помещения. Концепция электронного офиса позволяет существенно повысить эффективность работы служащих творческого труда, требующей интенсивной обработки больших объемов сложноорганизованной информации. Электронный офис позволяет осуществлять *е-процессы*, т. е. электронные процессы. Методика перехода от классических к *е-процессам*, а от них к *т-процессам* (к мобильным процессам) рассматривается в следующих разделах.

На рис.2.5.1 приведен список объектов, которые используются для построения диаграмм, описывающих офисную деятельность. Для сравнения на рис.2.5.2 приведен аналогичный список объектов, которые применяются для построения моделей промышленных (производственных) процессов. Разница в десяти объектах (они расположены в пяти последних рядах ниже разделительной линии) показывает, что в промышленных процессах преобладают технические и технологические объекты, а в офисных процессах преобладают информационные носители, так как офис

– это предприятие создающее, перерабатывающее и использующее только информацию.

К видам деятельности, осуществляемым в офисе, необходимо в первую очередь отнести: обслуживание входных информационных потоков разнообразных видов (офис-менеджмент), собственно документооборот и организацию делопроизводства, планирование и контроль выполнения производственных заданий, стратегическое и тактическое планирование основной деятельности, принятие решений разных уровней, техническую поддержку деятельности и др. (рис.2.5.3).

Основные функции и средства электронного офиса следующие: общая обработка документов и их оформление; хранение документов; обеспечение сквозного доступа к документам без их дублирования на бумаге; дистанционная и совместная работа служащих над документом; поддержка способов общения, не покидая рабочего места; электронная почта; персональная обработка данных; составление документов, воспроизведение и полиграфическое размножение документов; обмен информацией между БД; ввод данных или форм и ведение персональных БД; генерация отчетов по обработке данных; управление ресурсами; контроль исполнения; управление личным временем; контроль автоматической корреспонденции; поддержка технического и профессионального инструктажа служащих; передача данных; обеспечение разнообразия в наглядном представлении материала; обеспечение стилистического качества документов; моделирование решений и имитация их принятия; информационная поддержка принятия решений; работа со средствами автоматизированного обучения; служба консультаций; создание адаптируемых автоматизированных рабочих мест; обмен локальной и персонализированной информацией; служба видеотекста; обмен и интеграция программных средств; перенос документов с одного носителя на другой; полиграфическое оформление документов; телефонные и телевизионные совещания; групповой контакт через терминалы и др. Общим свойством и задачей офисных ИС является поддержка документооборота. Этот и другие виды работ по поддержке рутинных операций с документами составляют самостоятельный класс функций, характерных для офиса (что также справедливо для соответствующих средств автоматизации). Значительный экономический эффект от деятельности организации достигается за счет внедрения научной организации труда и повышения эффективности офисных процессов на базе современных информационных технологий. На рис.2.5.4 – 2.5.5 приведены функциональные деревья, структурирующие перечисленные офисные процессы.

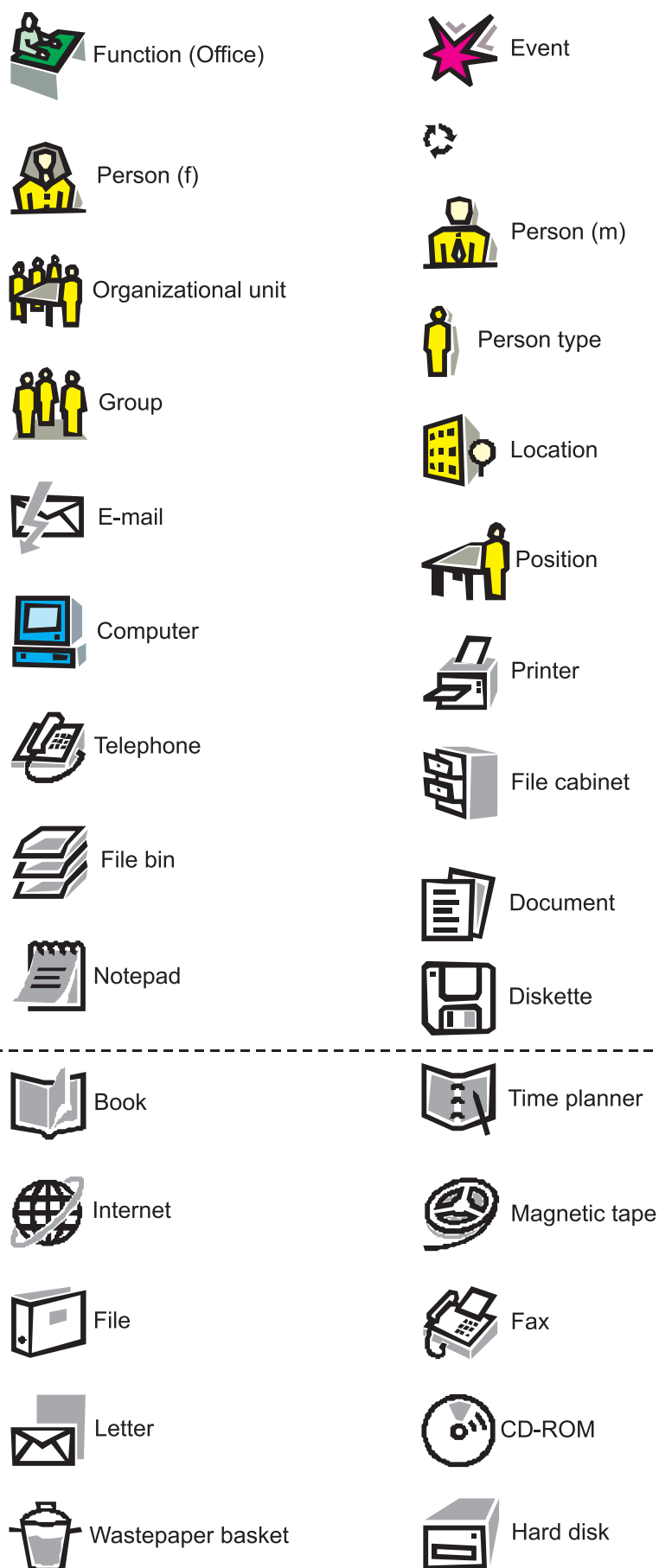


Рис.2.5.1 Перечень символов объектов, используемых в моделях типа Office process

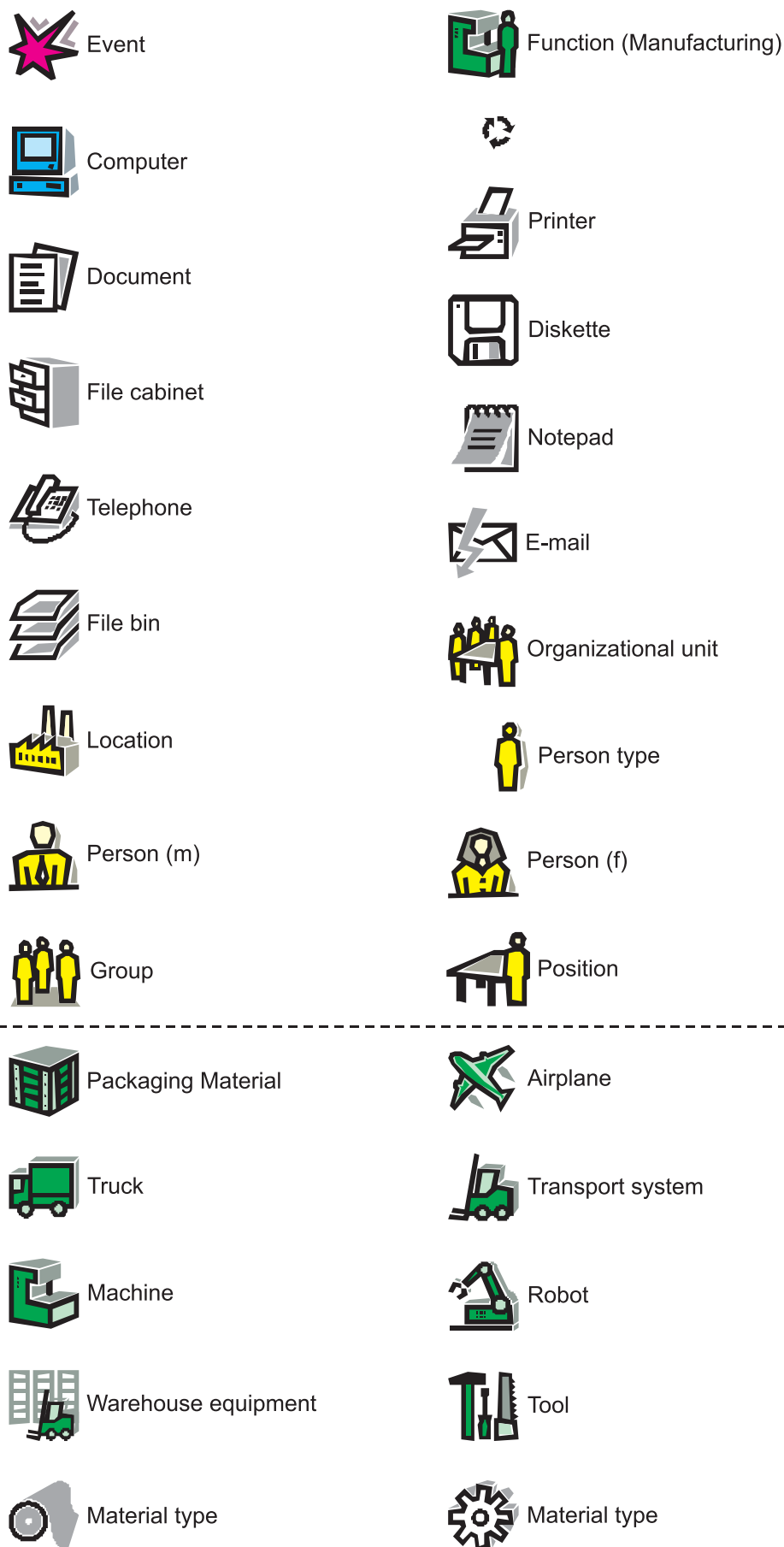


Рис.2.5.2 Перечень символов объектов, используемых в моделях типа Industrial process

Функциональные деревья (Function tree) содержат один тип объекта (функцию), между экземплярами которой допустим (для моделей функционального вида с простым фильтром) один тип связи (является процессно-ориентированным вышестоящим).

## 2.6. Распределенные информационные системы

Объединение распределенных на большой территории разнообразных компьютерных сетей, обеспечение взаимодействия различных компьютеров и интеграция ИС осуществляются по различным технологиям, например, по технологии «клиент-сервер», технологии совместного использования ресурсов в рамках глобальных сетей, технологии универсального пользовательского общения в виде электронной почты. Наиболее распространена технология «клиент-сервер», где обычно один компьютер в сети располагает информационно-вычислительными ресурсами (мощный процессор, файловая система, почтовая служба, служба печати, БД и др.), а другие компьютеры пользуются ими. Компьютер, управляющий тем или иным ресурсом, называют *сервером* этого ресурса, а компьютер, желающий им воспользоваться, – *клиентом*. Если ресурсом являются БД, то говорят о *сервере БД*, назначение которого – обслуживать запросы клиентов, связанные с обработкой данных; если ресурс – файловая система, то говорят о *файл-сервере*, и т. д.

Реализации технологии «клиент-сервер» в конкретных программных системах могут существенно различаться, но принципы у них общие. Один из основных принципов заключается в разделении операций обработки данных на три группы. В первую входит ввод и отображение данных. Во вторую – прикладные операции обработки данных, характерные для решения задач предметной области. В третью группу входят операции хранения и управления данными (БД, файловыми системами, ИС). В соответствии с этим выделяют три модели реализации технологии «клиент-сервер».

В AS-модели, т. е. модели сервера приложений (Application Server), программа, выполняемая на компьютере-клиенте, решает задачу ввода и отображения данных, т. е. реализует операции первой группы. Прикладные программы выполняются одним либо группой серверов приложений (один или несколько удаленных компьютеров). Прикладные программы обеспечивают доступ к ресурсам различных типов (БД, индексированным файлам, очередям и др.).

В AS-модели реализована трехзвенная схема раз-

деления операций, где прикладная программа выделена как важнейшая. Основным элементом этой схемы является сервер приложения. Он реализует несколько прикладных функций, каждая из которых оформлена как служба и предоставляет услуги всем программам, которые желают и могут ими воспользоваться. Серверов приложений может быть несколько, и каждый из них предоставляет определенный набор услуг. Любая программа, которая пользуется ими, рассматривается как клиент приложения.

В AS-модели используются мониторы обработки транзакций (Transaction Processing Monitor – TPM), или просто мониторы транзакций. Это программные системы, обеспечивающие эффективное управление информационно-вычислительными ресурсами в распределенной сети. Понятия транзакции в TPM и в традиционных СУБД несколько различны. Транзакция в СУБД – это атомарное действие над БД. В TPM транзакция трактуется гораздо *шире*: она включает не только операции с данными, но и любые другие действия – передачу сообщений, запись в индексированные файлы, опрос датчиков и т. д. Это и позволяет реализовывать в TPM прикладные действия предметной области (СУБД это не делает).

Основная функция TPM – обеспечить быструю обработку запросов, поступающих к AS от множества клиентов (сотен и даже тысяч). Эффективная обработка сообщений может быть повышена за счет использования систем управления очередями. Разработчики TPM обычно включают в арсенал своих систем специальный менеджер ресурсов, отвечающий за управление очередями. Управление очередями возложено на специальную программу. Помещение в очереди и выборка из них – прерогатива серверов, которые запрашивают менеджер очередей для выполнения соответствующих действий.

Упрощенно работа с очередями выглядит следующим образом. Пользователь посылает запрос конкретной службе (выделенному серверу), который помещает сообщение в очередь запросов к данной службе. Другой сервер извлекает сообщение из очереди запросов, выполняет предписанные действия и формирует ответ на запрос также в виде сообщения, посылая его в очередь ответов. Возможность хранения очередей сообщений в долговременной памяти позволяет говорить о практически стопроцентной надежности взаимодействия клиента и сервера. В случае сбоя компьютера все сообщения сохраняются, а их обработка возобновляется с той точки, где произошел сбой.

Уникальным симбиозом компьютеров и коммуника-

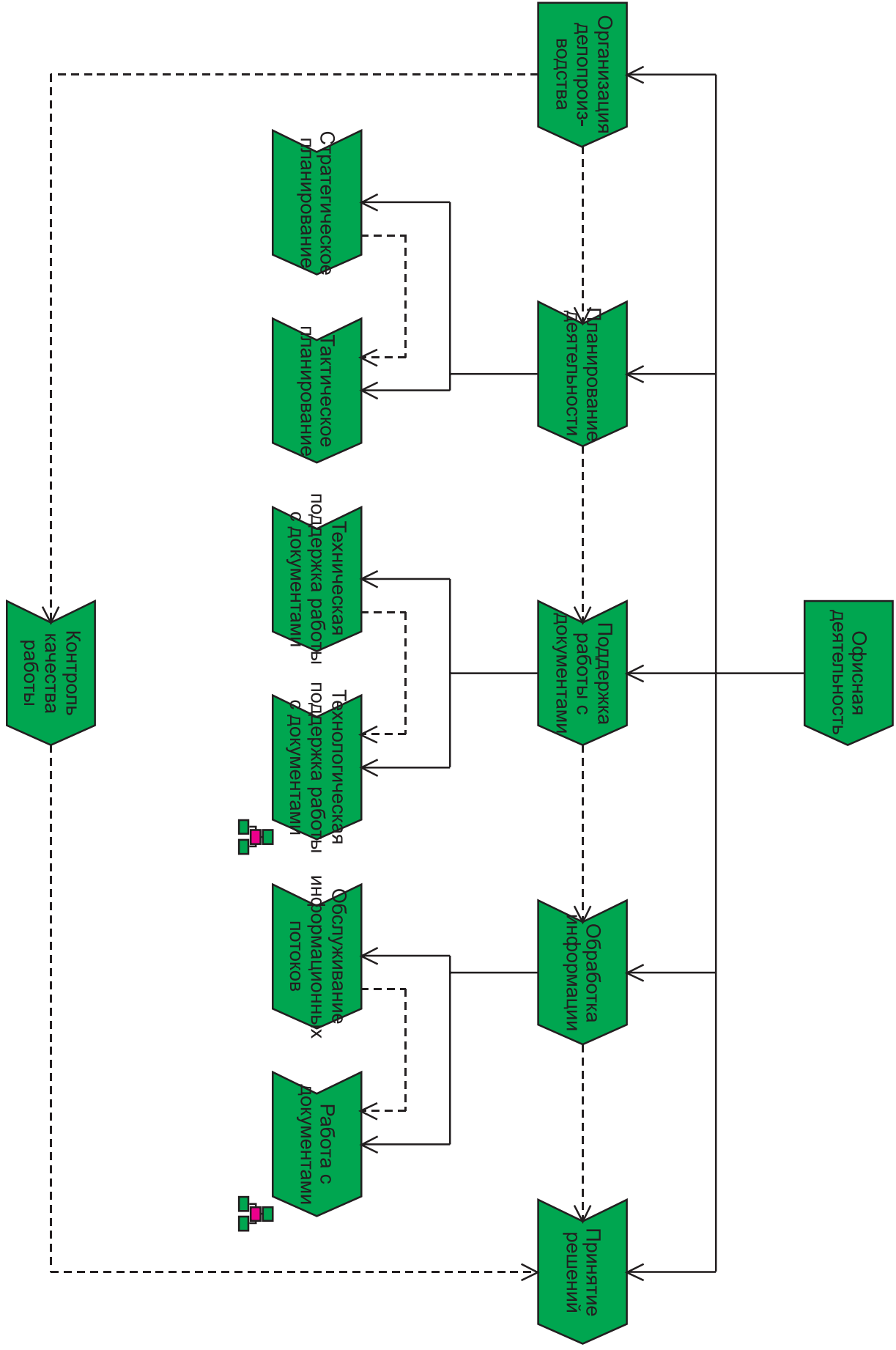


Рис.2.5.3 Основные процессы офисной деятельности



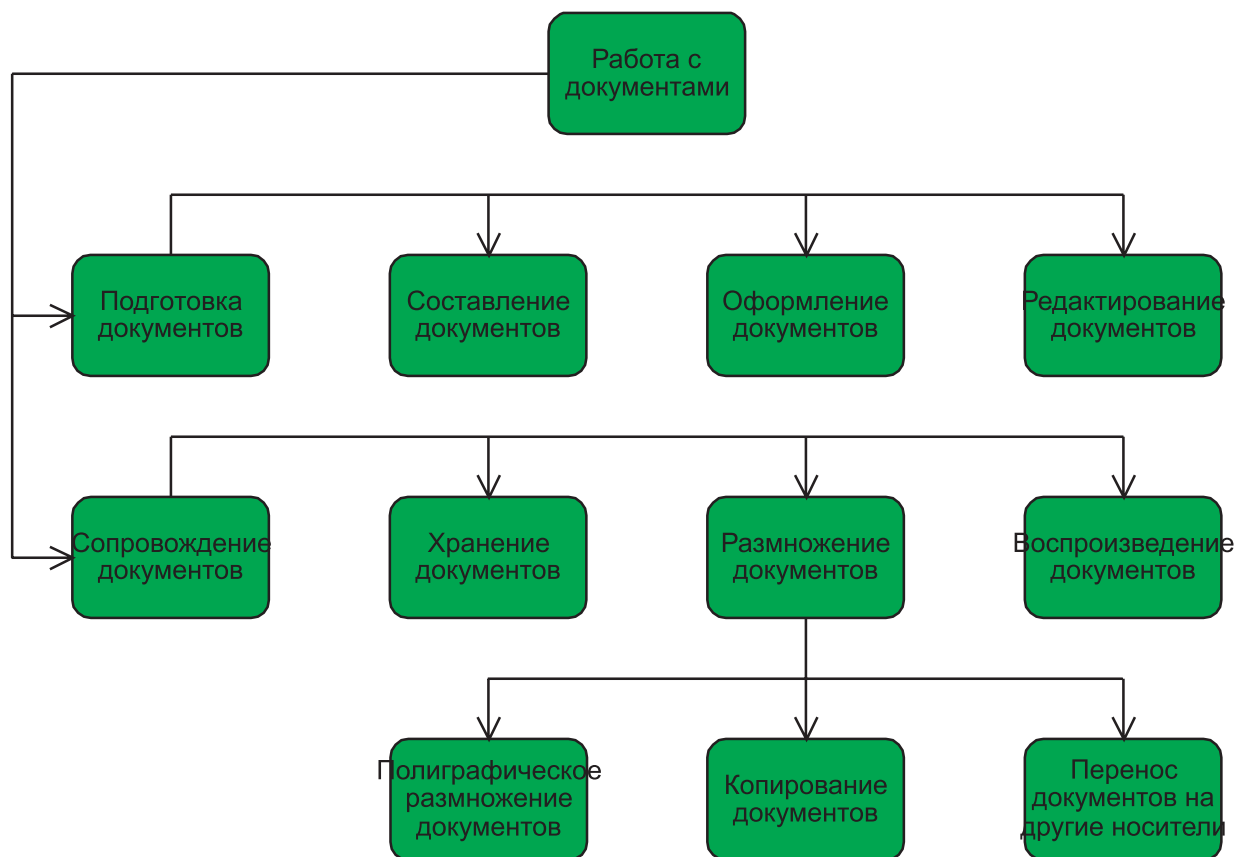


Рис.2.5.4 Фрагмент дерева функций, детализирующих процесс работы с документами

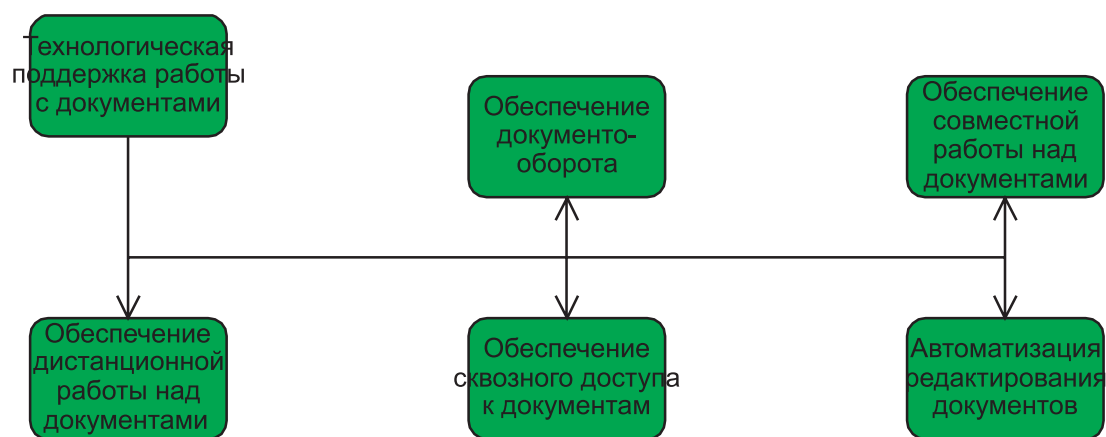


Рис.2.5.5 Детализация функции «Технологическая поддержка работы с документами»

ций являются *глобальные вычислительные и информационные сети*. С расширением компьютерных коммуникаций возрастает потребность в средствах структурирования, накопления, хранения, поиска и передачи информации. Удовлетворению этих потребностей служат информационные сети и их ресурсы. Совместное использование ресурсов сетей (библиотек программ, БД, ИС, вычислительных мощностей) обеспечивается технологическим комплексом и средствами доступа.

*Глобальные сети* – это *телекоммуникационные структуры, объединяющие локальные информационные сети, которые имеют общий протокол связи, методы подключения и протоколы обмена данными*. Каждая из глобальных сетей (Internet, Bitnet, DECnet и др.) организовывалась для определенных целей, а в дальнейшем расширялась за счет подключения локальных сетей, использующих ее услуги и ресурсы.

Крупнейшей глобальной информационной сетью является Internet. Передача данных в этой сети организована на основе протокола Internet – IP (Internet Protocol), представляющего собой описание работы сети, которое включает правила налаживания и поддержания связи в сети, обращения с IP-пакетами и их обработки, описания сетевых пакетов семейства IP. Сеть спроектирована таким образом, что пользователь не имеет никакой информации о конкретной структуре сети. Чтобы послать сообщение по сети, компьютер размещает данные в некий “конверт”, называемый, например, IP с указанием конкретного адреса.

Сеть может состоять из совершенно разнородных подсетей, соединенных друг с другом шлюзами. В качестве подсетей могут выступать локальные сети, национальные, региональные и специализированные сети, а также другие глобальные сети, например, Bitnet или Sprint. К этим сетям могут подключаться машины разных типов. Каждая из подсетей работает в соответствии со своими специфическими требованиями и имеет свою природу связи, сама разрешает свои внутренние проблемы. Однако предполагается, что подсеть может принять пакет информации и доставить его по указанному в этой подсети адресу. Таким образом, две машины, подключенные к одной подсети, могут напрямую обмениваться пакетами, а если возникает необходимость передать сообщение машине другой подсети, то вступают в силу межсетевые соглашения, для чего подсети используют межсетевой язык – протокол IP. Сообщение передается по цепочке шлюзов и подсетей, пока оно не достигнет нужной подсети, где доставляется непосредственно получателю.

Для обеспечения доступа к глобальным сетям

пользователю необходимо осуществить подключение к подсети, используя определенные методы доступа, основанные на взаимосвязи протокола обмена и типа линии связи. Виды доступа отличаются не только технологией, но и стоимостью.

Популярной услугой вычислительных сетей является *электронная почта*. Поэтому поставщики сетевых операционных систем комплектуют свои продукты средствами поддержки электронной почты.

Электронная почта в локальных сетях обеспечивает передачу документов и успешно используется при автоматизации офисных работ. При использовании для связи между сотрудниками всего офиса она оказывается удобнее телефона, так как позволяет передавать такую информацию, как отчеты, таблицы, диаграммы и рисунки, которые по телефону передать трудно.

Использование электронной почты, как правило, предполагает передачу сообщений через специальные “почтовые ящики”, между которыми размещаются устройства обработки данных. “Почтовый ящик” – общая область памяти вычислительной сети, предназначенная для записи информации с помощью одной прикладной программы с целью ее дальнейшего использования другими прикладными программами, функционирующими в других узлах сети.

Электронная почта глобальных сетей передачи сообщений, где могут объединяться компьютеры самых различных конфигураций и совместимостей, обеспечивает следующие функции. Во-первых, работу в офлайн-режиме, когда не требуется постоянного присутствия на почтовом узле. Достаточно указать специальной программе-почтовику время системных событий и адреса, где следует забирать почту. Во-вторых, доступ к телеконференциям. В-третьих, доступ к файловым телеконференциям. Файловые телеконференции отличаются от обычных тем, что в качестве сообщений в них существуют не письма, а файлы. Например, создается файловая телеконференция, посвященная экономике, где каждый участник конференции может поместить свой файл, а другие участники этот файл непременно получат.

Существуют и другие возможности, предоставляемые членам сети. Можно, например, послать заказ на посылку или прием факса. Составляется обычное электронное письмо, оформленное должным образом, и посылается на адрес компьютерного узла, занимающегося факсимильными операциями. Текст этого письма в виде факса будет доставлен на факсимильный аппарат адресата.

К преимуществам электронной почты относятся ско-

рость и надежность доставки корреспонденции, относительно низкая стоимость услуг, возможность быстро ознакомиться с сообщением широкий круг пользователей.

Любая система электронной почты состоит из двух главных подсистем. Во-первых, клиентского программного обеспечения, с которым непосредственно взаимодействует пользователь. Во-вторых, серверного программного обеспечения, которое управляет приемом сообщения от пользователя-отправителя, передачей сообщения, направлением сообщения в почтовый ящик адресата и его хранением в этом ящике до тех пор, пока пользователь-получатель его оттуда не достанет. Серверное программное обеспечение при совместимости протоколов передачи данных может обрабатывать почту, подготовленную различными клиентскими программами. Это программное обеспечение различается уровнями производительности, надежности, совместимости, устойчивостью к ошибкам, возможностями расширения.

Клиентское программное обеспечение предоставляет пользователям удобные средства для работы с почтой. Несмотря на их многообразие в различных системах электронной почты, все они имеют общие функции: оповещение о прибытии новой почты, чтение входящей почты, создание исходящей почты, адресация сообщений, использование адресной книги, содержащей список абонентов, которым часто посылают почту,

отправка сообщений, обработка сообщений и их сохранение. К обработке сообщений относятся такие функции, как печать, удаление, переадресация письма, сортировка, архивирование сообщений, хранение связанных сообщений. Особо следует выделить программы, позволяющие работать с папками, создавать свои папки для хранения в них сообщений по различным темам. Это очень удобно и помогает быстрее и эффективнее обрабатывать почту.

Различные почтовые программы могут быть классифицированы по разным признакам. Например, в какой операционной системе они могут работать. Наиболее массовое распространение получили продукты, работающие в Windows. Другим важным признаком классификации является функциональная возможность почтовых программ, например, обработка мультимедийных сообщений, возможность работы с разными кодировками сообщений, наличие многопользовательского интерфейса и др. К дополнительным признакам можно отнести: интерфейс пользователя, качество справочной системы, интеграция с другими пакетами, требуемое дисковое пространство для установки и цена.

Рекомендуем читателям самостоятельно составить ARIS-диаграммы, отражающие приведенное здесь текстовое описание некоторых аспектов распределенных ИС.

### 3. ПРОЦЕССНО-ОРИЕНТИРОВАННОЕ УПРАВЛЕНИЕ

В этой главе вводятся в обращение основные понятия процессной терминологии, сравнивается функционально- и процессно-ориентированное управление, а также кратко излагается переход к процессному управлению через этап усовершенствования (инжиниринга и реинжиниринга) бизнес-процессов.

#### 3.1. Функционально-ориентированное управление

В организации и управлении российскими предприятиями доминирует *структурный подход*. Структура (от лат. *structura* – строение, расположение, порядок) – совокупность устойчивых элементов сущности, обеспечивающих ее целостность. Организационную структуру можно определить как упорядоченную совокупность устойчиво взаимосвязанных элементов, обеспечивающих функционирование и развитие организации как единого целого. Структурный подход основан на иерархической *организационной структуре* предприятия, где организация и управление деятельностью предприятием осуществляется по структурным элементам (бюро, отделам, цехам и т. п.), а взаимодействие структурных элементов – через работу должностных лиц (начальников отделов, производств, цехов и т. д.).

Система управления практически всех российских предприятий имеет ярко выраженную иерархическую структуру и функциональную направленность. Есть Руководитель, ему подчиняются Замы по направлениям, которым в свою очередь подчиняются руководители подразделений, выполняющие определенные функции, и т. д. В основе подобной организации управления лежит принцип разделения и специализации труда Адама Смита, сформулированный еще в XVIII веке. В современных условиях этот подход управления оказывается неэффективным по следующим причинам. Во-первых, в функционально-ориентированном управлении (ФОУ) каждый начальник (стоящий выше в иерархической структуре управления) делегирует определенные (часто им самим) функции своему подчиненному (на следующий уровень иерархии), и главным потребителем (контролером) результатов труда работника является его вышестоящий начальник. Поэтому каждый сознательно или подсознательно старается угодить начальнику, а не коллеге из соседнего подразделения, а тем более клиенту. При современных тенденциях клиентской ориентации, когда клиент – “царь и бог”, такой подход сразу отбрасывает предприятие на последние роли в конкурентной борьбе за доли рынка. Во-вторых, при ФОУ усложнен обмен информацией

между различными подразделениями, что приводит к большим накладным расходам, неоправданно длительным срокам выработки управленческих решений и, как следствие, потере клиентов. По подсчетам аналитиков время работы и взаимодействия между субъектами ответственности разделяется следующим образом: 20% – время работы, 80% – передача результатов следующему исполнителю. Попытки внедрения автоматизации при ФОУ (программа внедрения автоматизированных систем управления 70-х годов прошлого столетия) привели к увеличению накладных расходов на обеспечение управления деятельностью предприятия без повышения эффективности, а в ряде случаев – при снижении эффективности в несколько раз. В-третьих, ФОУ противоречит фундаментальному закону искажения информации при ее передаче с помощью естественного языка (закону информационной энтропии). Любой естественный язык обладает информационной избыточностью, что является источником искажения сути сообщения. Поэтому при передаче информации через четыре уровня иерархии мы имеем высокую вероятность получения 100%-х искажений от исходного сообщения. Отчасти по вышеперечисленным причинам распались многие великие иерархические империи от древнейших времен до сегодняшних дней.

В организации и управлении деятельностью предприятия выделяют (кроме уже указанных) следующие недостатки ФОУ: разбиение технологий выполнения работы на отдельные, как правило, не связанные между собой фрагменты, которые выполняются различными элементами организационной структуры; отсутствие цельного описания технологий выполнения работы, в лучшем случае существует только фрагментарная (на уровне структурных элементов), и то не совсем актуальная документируемость технологий; отсутствие ответственного за конечный результат и контроля над технологией в целом; отсутствие ориентации на внешнего клиента, а также внутренних потребителей промежуточных результатов деятельности; высокие накладные расходы, как правило, непонятно откуда появляющиеся; неэффективность информационной поддержки, обусловленная “лоскутной” автоматизацией дея-

тельности отдельных подразделений и неудачными попытками внедрения корпоративных ИС.

Практика показывает, что эффективная деятельность в бизнесе (т. е. деятельность, приносящая максимальную прибыль) не может осуществляться при ФОУ и структурном подходе, где управление реализуется посредством приказов, распоряжений и разрешений. Одним из выходов является переход к процессно-ориентированному управлению.

### 3.2. Бизнес-процессы

Информационные системы играют важнейшую роль при клиентно-ориентированной организации бизнеса и процессно-ориентированном управлении бизнесом. Для выяснения этой роли введем несколько понятий из “процессной” терминологии.

**Бизнес-процесс** (БП) – это взаимосвязанный набор действий (операций, функций), которые по определенным правилам преобразуют исходные экономические ресурсы в конечные продукты или услуги. Под БП понимают совокупность различных видов деятельности, которые вместе взятые создают результат (продукт, услугу), имеющий ценность для потребителя, клиента или заказчика. БП выполняется с целью создания выхода (результата), спрос на который предъявляют корпоративные (внутренние) или внешние потребители (клиенты). В качестве клиента может выступать другой БП.

Можно сказать, что БП – это цепочка работ, результатом которой является какой-либо продукт или услуга. В цепочку обычно входят операции, которые выполняются по определенным бизнес-правилам структурными элементами, расположенными на различных иерархических уровнях организационной структуры предприятия. Под *бизнес-правилами* понимают способы реализации функций (бизнес-функций) в рамках БП, а также характеристики и условия выполнения БП, т. е., другими словами, бизнес-правила – это специальные сведения и особенности конкретной бизнес-деятельности. Вся бизнес-деятельность должна быть проникнута заботой о потребителе, т. е. в каждом БП должны учитываться требования потребителя (качество, стоимость, условия доставки, особенности использования, сервис и т. д.).

Предприятия можно рассматривать как экономические подсистемы, состоящие из ограниченного множества БП, работающих вместе для достижения конечных целей. К БП относятся, например, процесс сбыта и снабжения, процесс разработки нового изделия и вывода его на рынок, процесс обслуживания клиентов. Лозунг нефтяных компаний “От скважины до бензоа-

правки” означает БП макроуровня, охватывающий весь технологический цикл компании. БП “Материально-техническое обеспечение промышленного предприятия” изучает целая научная дисциплина – “Промышленная логистика”.

Не существует стандартного списка БП. Каждое предприятие имеет или должно классифицировать и описать свой собственный перечень БП. Как правило, основу для классификации составляют четыре категории БП – *основные, обеспечивающие, процессы управления и развития*.

Основными являются те БП, которые создают основные выходы (продукцию или услуги), представляющие ценность для клиента и приносящие основной доход организации. Как правило, основных БП в организации немного, обычно не более десяти.

Обеспечивающими являются БП, которые обеспечивают выполнение основных БП. В общем случае они обеспечивают ресурсами все БП. В отличие от основных количество обеспечивающих БП достигает десятков.

БП управления охватывают весь комплекс функций управления на уровне каждого БП и организации в целом с учетом полного цикла управления организацией, начиная от стратегического планирования до анализа причин отклонения от плана и формирования управляющих решений.

К БП развития, как правило, относятся БП совершенствования выхода (производимого продукта или услуги) или технологии его получения, а также инновационные процессы.

Кроме основных БП (они приносят основной доход организации), можно выделить *не основные БП*, которые приносят незначительную долю дохода. Более того, в составе основного БП могут существовать внутренние обеспечивающие подпроцессы, а в составе обеспечивающих можно выделить основной обеспечивающий и несколько вспомогательных подпроцессов. Обеспечивающие процессы снабжают ресурсами и технологически поддерживают основные БП, а также процессы управления и развития.

К общим обеспечивающим БП относятся, например, формирование спроса на продукцию или услуги, экономический анализ, финансовое обеспечение, снабжение, юридическое обеспечение, обеспечение транспортом, ремонт оборудования, обеспечение теплом (электроэнергией), обеспечение персоналом, обеспечение охраны труда и техники безопасности, социальное обслуживание, охрана объектов и материальных ценностей.

К процессам развития относят, например, расшире-



ние производственных площадей, совершенствование технологий производства или предоставления услуг, внедрение нового оборудования, разработку или внедрение ИС.

Каждый БП характеризуется четко определенными во времени началом и концом; внешними интерфейсами, которые либо связывают его с другими БП внутри организации, либо описывают выход во внешнюю среду; последовательностью выполнения функций и правилами их выполнения (бизнес-правилами). Для каждой функции, входящей в БП, определены ее место в общей последовательности работ, исполнитель, условия инициации (события), время и стоимость выполнения и др.

БП можно представить как упорядоченную совокупность таких сущностей, как рабочие объекты, ресурсы, организационные единицы, функции, события и др. Рассмотрим краткое описание перечисленных составляющих БП.

*Рабочий объект* – это сущность, над которой осуществляется некоторое действие (преобразование, обработка, формирование). Рабочие объекты могут быть или материальными (сырье, материалы, полуфабрикаты, готовые изделия), или финансовыми (платежи, перечисления и др.), или информационными (заказы, накладные, счета и другие документы). Рабочие объекты являются динамическими сущностями, т. е. периодически возникают в БП (закупаются, производятся, формируются), преобразуются в другие объекты и удаляются из БП (продаются, передаются на хранение). Под рабочими объектами можно понимать предметы труда.

*Ресурсы* – это сущности, с помощью которых осуществляются БП, например, оборудование, деньги на расчетном счете, нормативно-справочная информация (номенклатуры-ценники, классификаторы) и т. д. Ресурсы, так же как и рабочие объекты, могут динамически изменять свое состояние (конкретные экземпляры ресурсов в течение жизненного цикла экономической системы добавляются или удаляются), но в отличие от рабочих объектов используются в системе в течение нескольких циклов воспроизводства, т. е. находятся в системе постоянно и в этом смысле являются статическими. Под ресурсами можно понимать средства труда. Разделение сущностей на рабочие объекты и ресурсы зависит от использования их в том или ином БП, поскольку в одном БП сущность может выступать в роли рабочего объекта (например, предприятие самостоятельно ремонтирует некоторое оборудование), а в другом эта сущность используется в качестве ресурса (оборудование используется в производственном процессе).

*Организационные единицы* (предприятия, подразделения, персонал, отдельные исполнители) – это частный случай ресурсов, представляющих собой объединение исполнителей, которые используют другие ресурсы для выполнения БП. Одно и то же подразделение может участвовать в нескольких БП. Например, отдел сбыта участвует собственно в сбыте товаров, в формировании производственных заказов и предоставляет информацию в маркетинговый отдел для проведения исследований рынка, выработки стратегических решений, представляет учетную информацию для управления финансовыми ресурсами. И наоборот, в одном и том же БП принимают участие множество организационных единиц: например, в подборе и управлении кадрами принимают участие не только отдел кадров, но и производственные подразделения, финансовый отдел, дирекция предприятия.

*Функция или бизнес-функция (работа, действие, операция)* преобразует входные рабочие объекты в выходные или модифицирует их. Последовательность взаимосвязанных по входам и выходам функций составляет БП. Функция БП может порождать рабочие объекты любой природы (материальные, денежные, информационные).

По степени информатизации выделяют автоматические, интерактивные, экспертные и неавтоматизированные функции. *Автоматические функции* выполняются ЭВМ без участия человека, например, составление стандартных отчетов или проведение математических расчетов. *Интерактивные функции* выполняются ЭВМ и человеком в диалоговом режиме, например, реализация нестандартных запросов. *Экспертные функции* выполняются человеком на основе рекомендаций (команд), подготавливаемых ЭВМ. *Неавтоматизированные функции* выполняются человеком без использования ЭВМ.

*События* описывают различные условия протекания БП. Эти условия связаны с происходящими событиями во внешней среде или в самих БП (например, заказ принят, отвергнут, отправлен на корректировку), которые изменяют состояния объектов, ресурсов, организационных единиц. Таким образом, событие фиксирует факт завершения выполнения некоторой функции и перехода объекта в новое состояние или появления нового объекта. Новые состояния объектов вызывают выполнение новых функций, которые создают новые события и т. д., пока не будет завершен некоторый БП. Таким образом, последовательность событий составляет конкретную реализацию БП.

Каждое событие описывается с двух точек зрения:



информационной и процедурной. Информационно событие отражается в виде некоторого сообщения, фиксирующего сам факт выполнения некоторой функции, изменения состояния или появления нового объекта. Процедурно событие вызывает выполнение других функций и поэтому для каждого состояния объекта должны быть заданы описания вызовов других функций. Таким образом, события выступают в управляющей роли для выполнения функций БП и определяют направления материальных, информационных и финансовых потоков в зависимости от конкретной ситуации.

У каждого БП есть границы, интерфейс и владелец. *Границы БП* – граница входа, которая предшествует первой операции процесса, и граница выхода, которая следует за его последней операцией. *Интерфейс БП* – организационный, информационный и технический механизм, посредством которого БП взаимодействует с предшествующим и последующим БП. *Владелец БП* – лицо или группа лиц, отвечающие за БП и имеющие полномочия изменять его с целью усовершенствования.

Выделение БП, их описание, анализ и оптимизация – колоссальный резерв для повышения конкурентоспособности и эффективности работы компании. При совершенствовании БП на предприятии преследуют различные цели, например, стремятся улучшить их организацию, согласовать выполняемые работы, минимизировать используемые ресурсы (уменьшить затраты), ускорить работу и т. п. Цели совершенствования БП можно строго сформулировать. Например, возможны следующие формулировки целей совершенствования БП:

- сократить продолжительность БП “Разработка продукта” на 30 %,
- сократить время обслуживания клиентов до 5-ти минут,
- упростить управленческий учет и ускорить доступ к критически важной информации.

На рис.3.2.1 изображена диаграмма распределения функции, где вокруг произвольной бизнес-функции распределены различные сущности. Здесь в первом ряду расположено несколько организационных единиц, а в последних двух приведены различные информационные носители и т. д.

Под реорганизацией деятельности российских предприятий часто понимается внедрение новых средств автоматизации и информационных технологий, например, КИС или отдельных ее частей (электронных архивов, средств поиска документов, систем документооборота и т. д.). При несовершенных БП это тупиковый путь, так как после автоматизации

беспорядка получится автоматизированный беспорядок. Необходимо прежде всего заострить внимание на вопросах совершенствования БП предприятия, повышения их эффективности и оптимизации в том или ином смысле (в смысле быстродействия, минимизации энергозатрат и т. д.). Например, можно стремиться к построению оптимальных схем товарно-финансовых потоков и связанных с ними документооборотом.

Различают постепенное (инжиниринг) и кардинальное (реинжиниринг) совершенствования БП. Оба подхода основаны на анализе БП, который позволяет выявлять дублирование функций, узкие места, затратные центры, качество выполнения отдельных операций, избыточные операции, отсутствие или неполноту информации, возможности автоматизации, возможности внедрения систем управления качеством и многое другое. При анализе можно оценивать БП, например, путем пооперационного определения стоимости их выполнения. Естественно стремиться к уменьшению стоимости БП при сохранении необходимого качества.

### 3.3. Управление бизнес-процессами

Клиент оплачивает (или должен оплачивать) не структуру организации, а выходы процессов, в ней исходящих, т. е. товары или услуги. Эффективны те организации, в которых структура максимально соответствует выполняемым БП, т. к. последние подчинены целям существования (функционирования) организации, и структура организации должна соответствовать ее целям. БП, протекающие в организации, первичны по отношению к структуре этой организации (БП – первичны, структура – вторична).

Процессный подход ориентирован, в первую очередь, не на организационную структуру предприятия, а на БП. В этом подходе, во-первых, предприятие рассматривается как *бизнес-система*, которая представляет собой связанное множество БП, во-вторых, система управления предприятием ориентирована на управление как каждым БП в отдельности, так и всеми БП предприятия, в-третьих, система качества предприятия ориентирована на обеспечение качества выполнения БП.

При этом к БП выдвигаются следующие основные требования. БП должны соединять задачи предприятия в единое целое от начала до конца и ориентироваться на потребителя. БП должны пересекать вертикальные подразделения и способствовать “горизонтальному” видению организации. Необходимо, чтобы каждый работник организации знал об этих БП, имел какое-то представление о них. Это прежде всего необ-

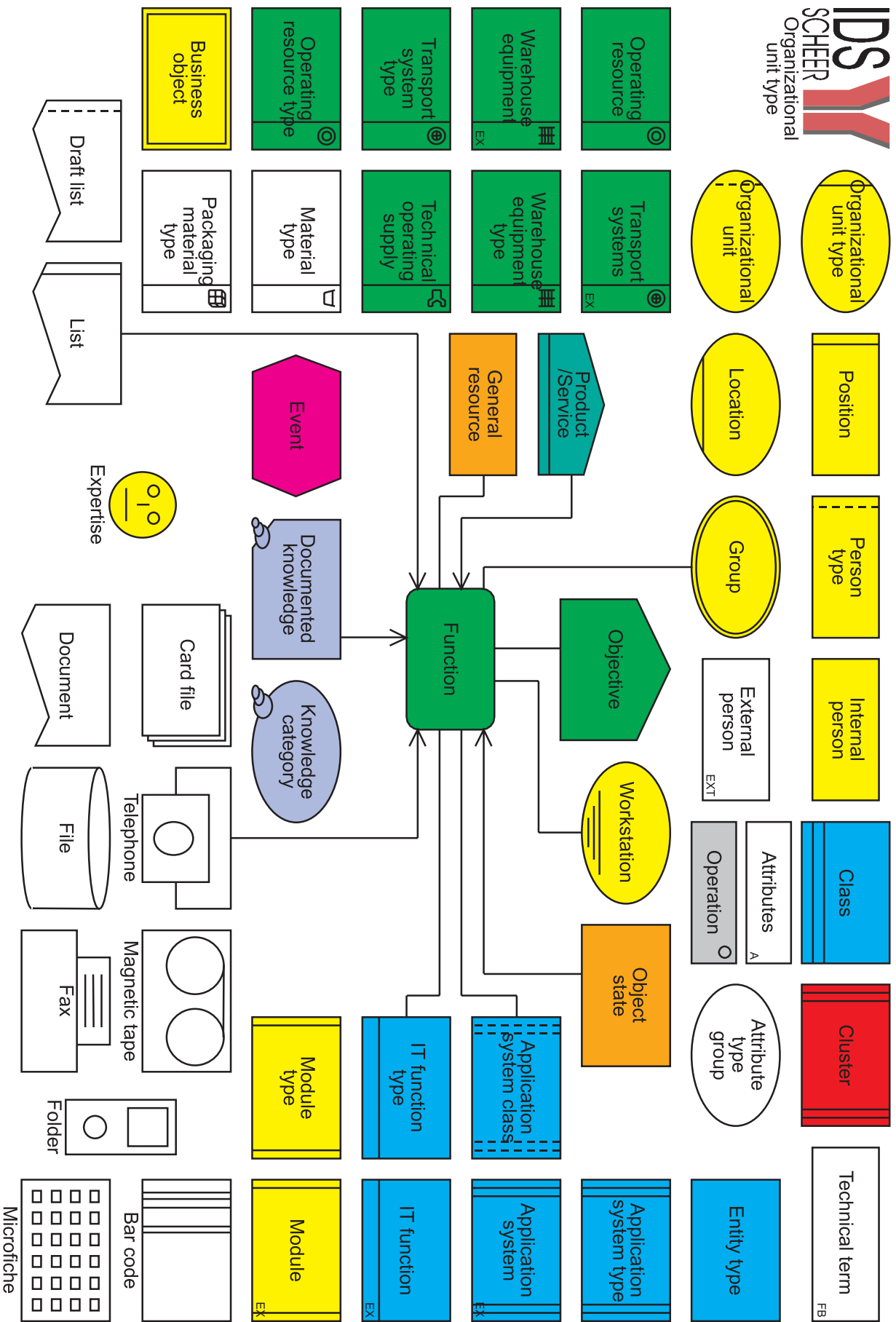


Рис.3.2.1 Диаграмма размещения (окружения) функции

ходимо для изменения установок работников. У каждого из них должна сложиться целостная картина работ. Несмотря на то, что каких-то БП его работа прямо не касается, он должен иметь о них представление и думать о своей работе с позиций целого как о его части.

Нужно разработать систему измерения БП. Все показатели должны быть увязаны в единую систему и отражать БП в целом. Не должно быть “показателей в себе”. Хорошо продуманные показатели не только позволяют оценивать и корректировать ход БП, они очень важны в организации командной работы, превращая группу разрозненных людей в единую команду, объединенную общей целью.

Идея представления предприятия в виде набора БП, а управление его деятельностью как управление БП стала распространяться в конце 80-х годов прошлого столетия. Лучшие компании мира на практике доказали эффективность, экономичность и прогрессивность перехода на клиентно-ориентированное производство (ориентированного на реального рыночного клиента) и процессно-ориентированное управление предприятием. Эти компании перешли на “тощее производство” или “тощую” *ресурсосберегающую организационную структуру*. Для этого они выполнили, например, следующее: делегировали большие полномочия и ответственность исполнителям, сократили количество уровней принятия решения, автоматизировали технологии выполнения БП, сместили акцент с контроля качества конечного продукта (услуги) на обеспечение качества выполнения БП предприятия. В этих компаниях, во-первых, все БП четко описаны, документированы и оптимизированы, во-вторых, за них отвечают и ими управляют конкретные специалисты.

Основным условием перехода к процессно-ориентированному управлению является высокий уровень средств информационного обеспечения деятельности предприятия. Когда вычислительные средства поддержки информационной деятельности не были доступны, существование функционально-ориентированного подхода в управлении было не только оправданным, но и единственно возможным. При использовании средств поддержки информационной деятельности возникает возможность охватить всю систему целиком, рассмотрев составляющие ее БП как единое целое и введя оперативный контур управления на передний план. В этом случае человек (субъект ответственности) вовлечен в БП, т. е. подчиняется его законам и логике, а не другому человеку со своими слабостями и проблемами. В результате:

- система управления выравнивает требования ко

всем участникам БП, т. е. требует от них равного качества на всех участках, в каждый момент времени и во всех работах;

- система управления становится саморегулируемой с точки зрения качества;

- количество уровней управления сокращается до двух, так как управляющий (ответственный за выполнение) БП взаимодействует с управляющим предшествующего и последующего БП;

- численность управленческого персонала уменьшается по мере оптимизации и стабилизации БП;

- резко возрастает эффективность, устойчивость и управляемость БП;

- образуется гибкая и самонастраиваемая система управления.

Для успешного существования такой системы необходимо обеспечить оперативной информацией все БП (с технологической точки зрения) и всех участников БП (включая руководителей предприятия). Поэтому без соответствующей ИС невозможен переход на клиентно-ориентированную организацию предприятия и его процессно-ориентированное управление.

Можно выделить следующие общие принципы процессного управления: разделение управления работой и управления работником, кооперация и командная работа, ориентация на целое, профессионализация и широкий диапазон знаний и умений, накопление и распространение знаний и методов, высокая степень самостоятельности и инициативы снизу, горизонтальное построение организации, акцент на динамику и качественную специфику интересов конечных потребителей. Соблюдение этих требований на практике может помочь в реализации процессного управления.

Заметим, что в результате частых и множественных изменений в организации меняется процессная ориентация, и организация становится трудно управляемой. В этой ситуации желательно упорядочить деятельность организации, благодаря созданию референтной модели, общего глоссария, полного перечня БП, регламентов на них и т. д.

### 3.4. Управление потоками работ

Управление системой возможно только через управление ее процессами. От качества управления процессами зависит эффективность существования системы в целом. Но и сам управляемый БП должен быть качественным, т. е. быстрым, мало затратным, удобным и т. д. Дефектный БП необходимо изменять. Но когда БП качественный, необходимо контролировать

его качественное выполнение. Для оценки качества управления как отдельным БП, так и группой БП используются различные показатели. Например, затраты на выполнение БП, временные характеристики БП и другие показатели качества выполнения БП.

Управлять всеми БП при помощи одной прикладной системы невозможно. Очень часто для этого требуется ряд прикладных систем для организации продаж, закупок, производства, учета и т. д. Например, общее управление БП осуществляет, как правило, отдельная система, называемая управлением потоками работ (workflow). Внедрение этой системы является явным признаком перехода на процессно-ориентированное управление предприятием.

Системы класса workflow, например, в офисных процессах передают обрабатываемые объекты (документы) с одного рабочего места на другое. В идеале эта передача осуществляется электронными средствами – от компьютера, установленного на одном рабочем месте, к компьютеру, где выполняется следующая операция. Для этого требуется детальное описание процедур применительно к конкретному БП и указание на соответствующих исполнителей.

По завершении той или иной операции система класса workflow, например, забирает исходящий документ из электронного ящика одного менеджера и пересылает его в электронный ящик для входящих сообщений на компьютере следующего менеджера. Системы workflow содержат информацию о состоянии процесса обработки, времени выполнения и пользователях каждого БП и при необходимости выдают данные для оценки стоимости и времени, а также предоставляют информацию для мониторинга БП.

Электронное связывание участников БП через технологию workflow позволяет оперативно доставлять результаты выполненной работы до следующих исполнителей, автоматически сигнализируя о конце предыдущей операции. Таким образом, сокращаются затраты времени на межоперационные переходы. При этом достигается более гибкое планирование и использование имеющихся ресурсов, которые становятся “тонкими агентами” БП.

Электронные коммуникации и безбумажная технология рабочих потоков позволяют сделать более прозрачным контроль исполнения БП, когда в каждый момент времени можно знать его состояние на конкретных рабочих местах. Отклонения в БП сразу становятся автоматически известными для всех заинтересованных исполнителей и менеджеров. Более того, развитые интеллектуальные ИС позволяют заблаговременно

осуществлять диагностику и прогнозирование развития БП. В этих условиях менеджерам нет необходимости осуществлять контроль исполнения БП посредством сбора бумажных отчетов. Мониторинг БП становится более оперативным, не требуется поддержание большого штата управляющего персонала среднего звена.

Кроме того, возможность оперативной связи участников БП с менеджерами в случае возникающих проблем посредством проведения теле- и видеоконференций также повышает оперативность принятия решений без потери времени на проведение традиционных совещаний. Принимаемые управленческие решения ментально становятся известными через электронную почту и доски объявлений для всех участников БП. Это также способствует сокращению числа уровней управления на предприятии.

### ***3.5. Этапы перехода к процессному управлению***

Для большинства российских предприятий переход на процессно-ориентированное управление (ПОУ) необходимо начинать с этапов наведения порядка в их деятельности, усовершенствования БП и внедрения корпоративной информационной системы (КИС). Первые два этапа часто игнорируются, и, например, после быстрого обследования предприятия заинтересованные лица лоббируют одну из существующих коммерческих КИС под красивыми лозунгами ее настройки и адаптации под конкретное предприятие. Но стоимость такой настройки может на порядок превышать стоимость самой КИС и требовать серьезных временных затрат, совместимых с затратами на разработку новой системы. При этом, как правило, еще до проведения обследования (да и вообще до появления платежеспособного заказчика) исполнители знают, какую именно систему они будут внедрять и осуществлять “адаптацию” результатов обследования. Другая крайность, т. е. детальное обследование предприятия и разработка на его основе собственной КИС, отражающей существующую на предприятии оргструктуру и технологии, только усугубляет ситуацию (информатизируя беспорядок, можно получить только “информационный беспорядок”).

Предприятие – это сложная организация, функционирование которой одному человеку понять зачастую просто невозможно. Руководство в общих чертах представляет себе общий ход дел, а исполнители досконально изучили только свою деятельность и уяснили свою роль в сложившейся системе деловых взаимоотно-

ношений. Но как предприятие функционирует в целом, не знает, как правило, никто. Поэтому для того, чтобы иметь предмет анализа, необходимо, в первую очередь, описать деятельность предприятия. Причем форма описания должна быть понятна и удобна для анализа. Графические нотации и диаграммные представления моделей как раз и служат для облегчения понимания сложных систем. В результате анализа таких моделей выявляются пути усовершенствования деятельности предприятия за счет улучшения ее БП, применения КИС и автоматизации управления.

Совершенствование БП осуществляется на основе оценки их эффективности, где критериями оценки могут быть стоимостные и временные затраты на выполнение БП, наличие дублирующих или конфликтующих друг с другом БП, степень загруженности сотрудников и др. В процессе моделирования деятельности предприятия и усовершенствования БП должен быть получен проект, содержащий достаточно информации для реализации на его основе ПОУ предприятием в рамках бюджета выделенных ресурсов и времени.

Первым шагом на пути перехода к ПОУ является *предварительное изучение предприятия*. Оно начинается с обследования деятельности предприятия и может потребовать от нескольких дней до нескольких недель. В процессе обследования предприятия выявляются функциональные деятельности каждого из его подразделений и функциональные взаимодействия между ними, информационные потоки внутри подразделений и между ними, внешние по отношению к предприятию объекты и внешние информационные взаимодействия. По окончании обследования строится и согласуется с заказчиком предварительный вариант модели деятельности предприятия, детализированный до уровня основных БП. Результаты предварительного изучения рассматриваются руководством и служат, во-первых, для оценки преимуществ внедрения ПОУ, во-вторых, для принятия решения о необходимости и возможности детального изучения, в-третьих, для обоснования временных затрат и стоимости детального изучения.

*Детальное изучение* строится на фактах, выявленных во время предварительного изучения и проведения обследования предприятия. Оно предполагает более детальное документирование ограничений существующей системы управления, а также уточнение БП до уровня, необходимого для написания спецификаций новой (модернизированной) системы. На этом этапе осуществляется обработка результатов обследования и построение уточненных моделей деятельности предприятия двух видов – “Как есть” и “Как должно быть”.

Модели “как есть” представляют собой “снимок” положения дел на предприятии (организационная структура, взаимодействия подразделений, принятые технологии, автоматизированные и неавтоматизированные БП и т. д.) на момент обследования. Они позволяют понять, что делает и как функционирует предприятие, в том числе выявить недостатки и узкие места, а также сформулировать предложения по улучшению ситуации. Модели “как должно быть” отражают перспективные предложения руководства и сотрудников предприятия, экспертов и системных аналитиков и позволяют сформировать видение новых рациональных технологий работы предприятия.

Главным результатом детального изучения является построение *системного проекта (модели требований)*, являющегося первой фазой разработки ПОУ (именно фазой анализа требований к системе), на которой требования заказчика уточняются, формализуются и документируются. Системный проект строится на основе модели “как должно быть” и результатов обследования предприятия в части выявления требований к будущей системе.

Фактически на этапе *разработки системного проекта* дается ответ на вопрос: “Что должна делать будущая система?”. Именно здесь лежит ключ к успешному переходу к ПОУ. В практике, например, автоматизации сложных систем, известно много неудачных реализаций именно из-за неполноты и нечеткости определения системных требований. Системный проект позволяет следующее: описать, “увидеть” и скорректировать будущую систему до того, как она будет реализована физически, уменьшить затраты на разработку и внедрение системы, оценить разработку по времени и результатам, достичь взаимопонимания между всеми участниками работы (заказчиками, пользователями, разработчиками, программистами и т. д.), улучшить качество разрабатываемой системы.

Диаграммы моделей легко понять и обнаружить какие-либо несоответствия и ошибки в БП. Поэтому при презентации системного проекта аналитик должен быть готов услышать критические замечания. В результате презентации принимается решение о продолжении разработки или ее прекращении, а также устанавливается цена перехода к ПОУ. Поэтому аналитик должен создать несколько альтернативных моделей, имеющих разный набор преимуществ и предполагающих различные капиталовложения для их реализации.

На этапе анализа требований, во-первых, необходимо понять, что предполагается сделать, а во-вторых, задокументировать это, т. е. если требования не зафик-



сированы и не сделаны доступными для участников проекта, то они вроде бы и не существуют. При этом язык, на котором формулируются требования, должен быть достаточно прост и понятен заказчику. Модель требований в виде множества структурированных и взаимоувязанных диаграмм деятельности предприятия и является таким документом.

Приведем некоторые рекомендации по структурированию моделей деятельности. Во-первых, структурирование должно осуществляться в соответствии с БП предприятия, а не в соответствии с его оргштатной структурой. Именно БП представляют ценность для клиента, и именно их улучшением необходимо заниматься. Модель, основанная на оргштатной структуре, может продемонстрировать лишь хаос, царящий в организации, на ее основе возможно внести предложения только об изменении этой структуры. С другой стороны, модель, основанная на БП, содержит в себе оргштатную структуру предприятия. Во-вторых, в моделях должны быть отражены основные виды деятельности предприятия и их взаимосвязи. Например, для автотранспортного предприятия можно выделить следующие деятельности: эксплуатация автотранспорта, ремонт и техническое обслуживание, контроль безопасности, управление производством, обеспечивающая деятельность. Сложные виды деятельности необходимо детализировать. Так “Обеспечивающая деятельность” может включать в себя “Учет кадров”, “Бухгалтерский учет”, “Экономическое планирование”, “Материально-техническое снабжение”, “Складской учет” и т. п. В-третьих, каждая из деятельностей должна быть детализирована на БП. Например, деятельность “Учет кадров” включает в себя БП “Прием на работу”, “Увольнение” и т. д. Дальнейшая детализация БП осуществляется посредством бизнес-функций. Так БП “Прием на работу” содержит в себе функции “Прием заявления”, “Оформление приказа”, “Регистрация” и др.

Построенные модели являются не просто реализацией начальных этапов разработки системы, но и техническим заданием на последующие этапы. Они представляют собой самостоятельный отделяемый результат, имеющий большое практическое значение. В частности, модель “как есть” включает в себя существующие неавтоматизированные технологии на предприятии. Формальный анализ этой модели позволит выявить узкие места в БП и предложить рекомендации по их улучшению (независимо от того, предполагается на данном этапе переход к ПОУ или нет). Модели позволяют осуществлять автоматизированное и быстрое обучение новых работников конкретному направлению

деятельности предприятия (так как ее технология содержится в модели) с использованием диаграмм (известно, что “одна картинка стоит тысячи слов”). С помощью моделей можно осуществлять предварительное рассмотрение нового направления деятельности с целью выявления новых потоков данных, взаимодействующих подсистем и БП.

Системный проект полностью независим и отделяем от конкретных разработчиков, не требует сопровождения его создателями и может быть безболезненно передан другим лицам. Более того, если по каким-либо причинам предприятие не готово к реализации проекта, он может быть “положен на полку” до тех пор, пока в нем не возникнет необходимость.

После выбора системного проекта на основе выявленных и согласованных требований осуществляется разработка следующих предложений: составление перечня автоматизированных рабочих мест предприятия и способов взаимодействия между ними, анализ применимости существующих систем управления для решения требуемых задач и формирование рекомендаций по выбору таких систем, принятие решения о выборе конкретной (существующей) КИС или разработке собственной системы, разработка требований к техническим средствам, разработка требований к программным средствам, разработка предложений по этапам и срокам перехода к ПОУ.

На следующем этапе (проектирования) на основании принятых решений по переходу к ПОУ осуществляется преобразование системного проекта в технический проект (модель реализации). На этапе проектирования вырабатываются пути реализации моделей требований, которые порождены и зафиксированы при анализе. На этом этапе осуществляется построение модели реализации (или физической модели), демонстрирующей, как система будет удовлетворять предъявленным к ней требованиям (без технических подробностей). Здесь определяются параметры реализации на уровне типов. Зная типы подсистем и потенциальную аппаратную платформу, можно приступить к поиску коммерческих пакетов, которые удовлетворяют требованиям, выявленным и зафиксированным на этапе системного проектирования.

Системный проект позволяет выбрать КИС, наиболее полно подходящую конкретному предприятию, либо отвергнуть этот путь и приступить к разработке и реализации собственной системы. Рекомендуются следующие критерии выбора готовой системы: поддержка большинства БП, выявленных при анализе требований, поддержка концептуальной модели данных,



наличие высокоуровневых механизмов разработки для компенсации отсутствующих данных и функций, функционирование на различных аппаратных платформах, достаточные размеры внутренних таблиц, локализация. Помимо чисто технических критериев выбора, важную роль играют также деловые критерии, например, опыт внедрения и надежность продавца.

Отметим недостатки разработки собственной КИС по сравнению с покупкой готовой системы. Трудозатраты на создание собственной КИС огромны и составляют сотни и тысячи человеко-лет, стоимость разработки соизмерима со стоимостью готовой системы (а часто значительно превышает ее). Использование готовой системы менее рискованно, чем разработка собственной. Готовая система внедряется поэтапно и поэтому частично может быть доступна в рабочем режиме гораздо быстрее, чем собственная.

### **3.6 Инжиниринг и оптимизация бизнес-процессов**

Цель бизнес-инжиниринга заключается в поиске максимально эффективных БП. Инжиниринг БП – это не одноразовое мероприятие, а постоянная работа для тех, кто за них отвечает. Помимо идеи непрерывного эволюционного совершенствования БП, существует более революционная реинжиниринговая концепция, когда перед предприятием ставится задача начать все с нуля (по-новому).

Оба подхода имеют свои достоинства. В определенной ситуации, когда у предприятия появляется шанс радикально переосмыслить свою деятельность, внося в нее фундаментальные изменения, можно провести реинжиниринг. Но и после его завершения БП остаются в постоянном движении. Возникают новые организационные формы, появляются новые способы ведения бизнеса, которые можно взять за прототип, изобретаются новые технологии, наконец, приобретаются знания и опыт, связанные с недавно внедренными БП. Все это влечет за собой необходимость адаптации новых БП.

К основным целям улучшения деятельности предприятия можно отнести: уменьшение стоимости продукции и услуг, повышение качества производимой продукции и услуг, ускорение производства продукции и услуг, переход к “быстрому производству”. Последнее представляет собой объединение сервиса, качества, гибкости и стоимости. Быстрое производство является полной противоположностью массового производства, которое изобретено в начале прошлого столетия Генри Фордом. Тогда у потребителя не было выбора и экономия за счет

масштабов производства оказывала решающее влияние на принятие решений. Сегодня необходимо, прежде всего, учитывать весьма разнообразные требования клиентов. Поэтому основой деятельности предприятий является максимальное удовлетворение покупателей с различными требованиями. При “быстром производстве” особое ударение делается на гибкость деятельности предприятия, которая обеспечивает потребителю возможность заказа разнообразной продукции с быстрой доставкой и по приемлемым ценам.

Для совершенствования БП могут потребоваться изменения (модификации) функционального потока (например, объединение нескольких функций), управленческих решений, организационной структуры или обязанностей субъектов ответственности, используемых данных или информационного обеспечения.

При инжиниринге и непрерывной оптимизации следует четко документировать и сохранять БП. Это является обязательной предпосылкой для оценки БП на стадии анализа. В “хранилищах процессов” находятся корпоративные знания предприятия о БП и информация о процессах-прототипах. Здесь же можно собирать модели текущих, унаследованных и даже будущих БП, которые послужат фундаментом для улучшений в будущем.

Для синтеза оптимальных БП можно применять следующие вспомогательные средства инжиниринга: оценка БП, эталонное сравнение БП, имитационное моделирование и оценка качества БП.

Для того, чтобы построить БП, отвечающий конкретным целям, необходимо оценить БП с точки зрения этих целей. Если цели носят финансовый характер, например, “сократить стоимость БП на 30 %”, то необходимо установить связь между стоимостью и БП.

Обязательной предпосылкой для определения стоимости является наличие всеобъемлющей процедуры расчета. В основе пооперационного исчисления стоимости лежит принцип разбиения БП на элементарные подпроцессы (операции). Сначала определяется средняя стоимость одноразового выполнения каждого подпроцесса. Затем с помощью соответствующих коэффициентов вычисляется стоимость всего БП.

Для каждого подпроцесса можно определить ставки стоимости (например, стоимость типичного заказа на поставку, стоимость типичного напоминания об уплате и т. д.). Пооперационный стоимостный анализ позволяет распределить накладные расходы по БП и избежать маловразумительных единых ставок.

Сопоставление анализируемого БП с аналогичным БП называется эталонным сравнением. Эталонные БП предлагаются консалтинговыми компаниями.

Расхождение между характеристиками эталонного и анализируемого БП может подсказать, как лучше организовать у себя БП. Целевыми критериями при эталонном сравнении могут выступать финансовые, временные или совокупные показатели, например, стоимость БП, пропускная способность или величина выхода.

Перечислим *качественные и количественные критерии эталонного сравнения*.

*Качество* БП можно оценивать по следующим критериям: процент отходов, процент доработки и переделки, процент дефектных партий товара (число продуктов), процент возвратов, цена гарантийного обслуживания, наличие и достоверность информации.

*Продуктивность* отражают объем выхода продукции, выход продукции на единицу используемого ресурса, стоимость одной “доброкачественной” единицы изделия, число выполненных заказов на один час трудозатрат, добавленное качество на одного работника, стоимость функций, добавляющих качество, в процентах.

*Временные показатели* отражают процент доставок в установленные сроки, время цикла изготовления продукта, время цикла доставки, число доставок с задержкой, время, требуемое для переналадки, время, требуемое для проверки, процент непродуктивного времени.

*Удовлетворенность клиентов* оценивается, например, следующими критериями: число клиентов, возобновляющих заказ на покупку, индексы удовлетворенности клиентов, реально ожидаемый выход для клиента, рекомендации по приобретению, удобство для пользователя.

*Эффективность офисной работы* отражают: время, требуемое для обработки заказа, число препятствий для клиента, среднее число контактов на оформление одного заказа, число дефектов и требований доработки и переделки, число допустимых исключений, задержки отчетности в днях.

Оценка БП на основе результатов пооперационного исчисления стоимости и эталонного сравнения имеет важнейшее значение. Но для построения оптимального БП разрабатывается несколько альтернатив поведения “что – если”, которые анализируются с помощью метода имитационного моделирования, и среди них определяется наилучший. Анализ различных альтернатив в гипотетических ситуациях (“что – если”) осуществляется на существующей модели БП.

Для анализа можно выбирать альтернативные варианты БП, различающиеся по таким аспектам как последовательность выполнения функций, время выполнения функций, характер поведения соответствующих организационных единиц. Альтернативы вырабатываются индивидуально на основе эмпирических данных либо

автоматическим способом – по случайному принципу.

### ***3.7. Основные направления реинжиниринга***

Основные проблемы, с которыми сталкиваются предприятия, являются следствием плохих БП, а не плохой оргструктуры. Реорганизация, т. е. переход от модели “как есть” к модели “как должно быть”, не означает изменение оргштатной структуры предприятия, хотя этот процесс действительно может отразиться и на структуре. Пытаться совместить новую оргштатную структуру со старыми БП и наоборот – это все равно, что “переливать прокисшее вино в новые бутылки”.

Предприятия, которые начинают улучшать свою деятельность с уничтожения бюрократии, в результате получают хаос. Бюрократия – это тот цемент, который скрепляет все компоненты традиционной корпорации. Бюрократия решает проблему фрагментированных БП. Для того, чтобы избавиться от бюрократии и перестроить структуру предприятия, необходимо перепроектировать БП так, чтобы они не были фрагментарными. После этого предприятие может обойтись и без бюрократического аппарата.

Выбор нуждающихся в реорганизации БП состоит из следующих пяти этапов.

*Формулировка миссии предприятия*. Миссия должна объединять основную цель (задачу) и коренную причину существования предприятия. Миссия должна предоставить возможность отличать данное предприятие от его конкурентов.

*Выработка и согласование (на основе миссии) подцелей, вытекающих из главной цели*. Эти подцели получили название критических факторов успеха – SCF (Critical Success Factors). Рекомендуется иметь не более восьми SCF, в совокупности они должны отражать ключевые характеристики предприятия и его внешние связи. При этом каждый из них необходим, а все они вместе достаточны для достижения миссии. Другими словами, без достижения каждой из подцелей невозможно достижение главной цели, и, если все подцели достигнуты, то достигнута и главная цель.

*Выявление и согласование ключевых БП*. На данном этапе следует обратить внимание на следующие моменты. Общее число выбранных БП не должно превышать 15-20. Следует различать основные и вспомогательные БП. Основные процессы отражают целевую деятельность предприятия, т.е. производят товары (услуги), имеющие ценность для клиента: *ремонт и техническое обслуживание* на авторемонтном предприя-

тии, перевозка пассажиров в автобусном парке, продажа продуктов в магазине. Вспомогательные БП отражают обеспечивающую деятельность предприятия, большинство из них являются типовыми для предприятий различных отраслей (*Бухгалтерия, Учет кадров*), другие могут иметь серьезные отличия (*Обеспечение безопасности, Планирование*).

*Привязка БП к SCF.* На данном этапе необходимо соотнести БП и критические факторы успеха (какие процессы влияют на каждый из SCF?), для этой цели удобно построить таблицу “процессы-SCF”.

*Ранжирование и выбор БП для реорганизации.* Число SCF, на которые влияет БП, позволяет приблизительно оценить относительную важность данного БП. В результате оценки все БП ранжируются следующим образом: стратегически наиболее важные, но плохо работающие; менее влиятельные; минимально влияющие на работу предприятия или уже хорошо работающие.

Рассмотрим некоторые направления реинжиниринга, рекомендуемые Хаммером (Hammer) и Чампи (Champer) в книге “Reengineering the Corporation” [26].

*Отказ от массового контроля.* Контроль качества готового изделия является запоздалой и дорогостоящей мерой, неявно подразумевающей планирование брака. Качество готового товара невозможно улучшить, поэтому необходимо улучшать производство. Качество должно закладываться в изделие при его производстве. Каждый следующий заказ должен выполняться лучше предыдущего, необходимо постоянно совершенствовать материалы, умение и навыки сотрудников и т. п.

*Непрерывное обучение руководства.* Руководству желательно знать все БП предприятия, от появления исходных материалов до реализации готовой продукции потребителям. Хороший управляющий начинает свою работу в компании с низших звеньев. За несколько лет он проработает в разных подразделениях, и ему будут знакомы все проблемы предприятия.

*Переориентация руководителя от надзора к управлению.* Руководитель должен, прежде всего, знать работу, которую он контролирует, быть лидером и стремиться к повышению качества товаров и услуг. Он должен обладать реальной властью и быть ориентированным на информирование высшего руководства предприятия о проблемах и условиях, нуждающихся в изменении.

*Налаживание связей между подразделениями.* Сотрудники одного подразделения должны знать о проблемах, возникающих в смежных подразделениях. Фактически у каждого сотрудника есть свой потребитель в смежном подразделении: один, например, осуществляет закупку материалов, другой конструирует

изделия, источником брака может быть деятельность любого из них.

Чтобы научиться индуктивно мыслить во время реинжиниринга, необходимо гибче относиться к существующим правилам и использовать новейшие информационные технологии, их возможность опровергать те или иные правила и ограничения, принятые в традиционной работе. Ниже приведены примеры правил работы, которые могут быть отменены с применением новейших информационных технологий.

1) *Старое правило:* информация может появляться одновременно только в одно время и в одном месте. *Новая технология:* совместно используемые БД. *Новое правило:* информация может появляться одновременно во многих местах, если это требуется.

2) *Старое правило:* только опытные специалисты (эксперты) могут выполнять сложные работы. *Новая технология:* экспертные системы. *Новое правило:* менеджер может выполнять ту же работу, что и эксперт.

3) *Старое правило:* Приходится выбирать между централизацией и децентрализацией. *Новая технология:* телекоммуникационные сети. *Новое правило:* можно одновременно использовать выгоды, получаемые от централизации и децентрализации.

4) *Старое правило:* руководитель принимает все решения. *Новая технология:* средства поддержки принятия решений. *Новое правило:* принятие решений входит в обязанности каждого.

5) *Старое правило:* персонал нуждается в офисах для получения, хранения, коррекции и передачи информации. *Новая технология:* беспроводные средства коммуникации и портативные компьютеры. *Новое правило:* работники на местах могут получать и передавать информацию из любого места, где бы они ни находились.

6) *Старое правило:* лучший способ контакта с потенциальным покупателем – личный контакт. *Новая технология:* Internet. *Новое правило:* лучший способ контакта с потенциальным покупателем – эффективный контакт.

7) *Старое правило:* чтобы найти объект, необходимо знать, где он находится. *Новая технология:* автоматическое индексирование и отслеживание. *Новое правило:* объекты сами сообщают, где они находятся.

8) *Старое правило:* планы периодически пересматриваются. *Новая технология:* высокопроизводительные компьютеры. *Новое правило:* планы могут пересматриваться мгновенно и постоянно.

Из приведенных примеров очевидно, что по мере дальнейшего развития технологий будет происходить отказ от все большего количества правил, по которым организованы БП. Правила, которые представляются

непогрешимыми сегодня, могут устареть менее чем за год. Из этого вытекает, что использование возможностей, заложенных в новых технологиях, для изменений БП и резких его усовершенствований – это постоянная (а не одноразовая) деятельность. Следование новейшим технологиям и нахождение способов их применения на предприятии должно происходить непрерывно, так же, как исследования, разработки, маркетинг. Необходимо иметь практическое видение и творческую мысль для того, чтобы распознать возможности, скрытые в технологии, которая на первый взгляд кажется вообще не имеющей ничего общего с работой предприятия. Более того, предприятия должны сделать применение новых технологий одним из своих основных занятий, если они хотят быть впереди в наше время динамичных технологических изменений. Те, кто лучше сможет разглядеть и оценить возможности, скрытые в новой технологии, получают постоянное, растущее преимущество над конкурентами.

### **3.8. Характеристики перепроектированных бизнес-процессов**

Приведем из книги Хаммера и Чампи “Reengineering the Corporation” примеры общих свойств и характеристик перепроектированных БП.

*Несколько работ объединяются в одну.* Для перепроектированных БП характерно отсутствие сборочного конвейера: многие работы и задачи, которые раньше выполнялись по отдельности, теперь объединяются в одну, выполняемую одним специалистом (или, по крайней мере, специалистами одного подразделения, на которых возложена полная ответственность за выполнение работы). Безусловно, в таком подразделении могут возникнуть проблемы, связанные с распределением заданий и приводящие к задержкам и ошибкам, но они будут незначительными по сравнению с проблемами, возникавшими раньше, когда задания распределялись между разными подразделениями. Позитивным является то, что теперь каждый сотрудник знает, кто отвечает за быстрое и точное выполнение работы.

Выигрыш от введения интегрированных БП и отвечающих за них сотрудников может быть огромным. Предприятие избавляется от ошибок, задержек и дополнительной работы в связи с проблемами, которые возникали при распределении заданий. Обычно БП, ориентированный на такого сотрудника, выполняется в *десять* раз быстрее, чем его конвейерная версия. При этом, поскольку новый БП порождает меньшее

число ошибок и недоразумений, предприятию не нужны дополнительные работники для их исправления.

Кроме этого, интегрированный процесс требует меньше работы по его администрированию. Поскольку сотрудники, задействованные в процессе, отвечают за то, чтобы работа выполнялась вовремя и без ошибок, контроля за ними требуется меньше. Вместо этого, предприятие стимулирует этих работников, наделенных новыми полномочиями, к постоянному поиску новых творческих путей уменьшения длительности рабочего цикла и затрат при производстве качественного продукта или услуги. Еще одно преимущество интегрированных процессов – улучшение качества управления, поскольку в этих процессах занято меньшее число людей, становится легче давать им работу и следить за ее выполнением.

*Исполнители принимают решения.* Предприятия, осуществляющие инжиниринг, уплотняют БП не только горизонтально (возлагая на сотрудников множество последовательных заданий), но также и вертикально. Вертикальное уплотнение означает, что в тех местах БП, где сотрудники обычно обращались за ответом к руководству, теперь принимаются самостоятельные решения. В отличие от той ситуации, когда принятие решений изолируется от самой работы, в перепроектированном БП принятие решений становится *частью* работы, и теперь сами исполнители выполняют ту часть работы, которую раньше выполняло руководство.

*Этапы БП выполняются в естественном порядке.* Реинжиниринг БП освобождает их от предопределенной линейной последовательности выполнения этапов, при которой работы очередного этапа начинаются по завершению предыдущего. В перепроектированном БП этапы организованы в такой последовательности, в которой это необходимо. Отказ от линейности ускоряет работу БП по двум причинам. Во-первых, многие этапы выполняются одновременно. Во-вторых, уменьшается время между началом и окончанием выполнения БП и тем самым снижается вероятность перераспределения уже выполненной работы из-за устаревания информации или противоречий с ранее выполненными работами.

*Существуют различные версии БП.* Традиционные (стандартизированные) БП были ориентированы на производство массовой продукции для массового рынка. Все входные данные обрабатывались одинаково, и поэтому предприятия производили одинаковую продукцию. В настоящее время такая технология устарела: для того, чтобы удовлетворить современным требованиям, необходимо иметь несколько версий одного и того же БП,



каждая из которых была бы настроена на требования различных заказчиков, ситуаций и входных данных.

*Работа выполняется там, где ее целесообразно делать.* На структурно-ориентированных предприятиях работа выполняется специалистами, сгруппированными в структурные подразделения: бухгалтер умеет считать, а работник отдела снабжения умеет заказывать товары. Когда бухгалтерии нужны новые карандаши, приобретает их отдел снабжения. Снабженцы находят поставщиков, оговаривают цены, составляют заказ, проверяют поставленный товар и оплачивают счет. Это дорогостоящий БП, поскольку в его выполнении участвует несколько разных отделов, а также из-за дополнительной деятельности, связанной с обработкой всех нужных бумаг и связыванием воедино всех частей БП. Поэтому часто разумнее выполнить тот или иной вспомогательный БП в рамках основного БП под руководством и ответственностью владельца основного БП.

*Снижение доли работ по проверке и контролю.* Традиционные БП изобилуют операциями проверки и контроля, которые являются разновидностью бесполезной работы, поскольку не создают никаких ценностей, но тем не менее нужны для того, чтобы гарантировать отсутствие нарушений. Например, при выполнении типичного БП покупки отдел снабжения проверяет подпись заказавшего товар человека (чтобы убедиться, что он имеет на это право), а также имеется ли достаточно средств на счету данного подразделения. Все эти проверки имеют одну цель – убедиться, что подразделения не приобретают того, что они приобретать не должны. Возможно, эта цель достойна одобрения, но многие предприятия не замечают расходов, которые влечет организация строгого контроля. Все эти проверки требуют времени и сил – в некоторых случаях даже больших, чем сам БП “Приобретение товара” (более того, расходы на проверку могут оказаться выше стоимости приобретаемых товаров, например, как в случае с карандашами).

*Минимизация согласований.* Согласования являются другой разновидностью бесполезной работы, которую перепроектированный БП сводит к минимуму. Это достигается уменьшением числа точек внешнего контакта, имеющихся в БП, что приводит к снижению вероятности получения противоречивых данных, по которым, собственно, и требуется согласование.

*Ответственный менеджер является единственной точкой контакта.* Ответственный менеджер бывает полезен в ситуациях, когда различные этапы БП

настолько сложны или до такой степени разбросаны в пространстве, что одному человеку (и даже небольшой группе) не под силу справиться с их интеграцией. Выступая в качестве буфера между сложным БП и заказчиком, он ведет себя как отвечающий за выполнение всего БП, хотя на самом деле это и не так. Для того, чтобы выполнить эту функцию (то есть иметь возможность ответить на вопросы заказчика и разрешить возникшие у него проблемы), ответственный менеджер должен иметь доступ ко всем информационным системам, используемым сотрудниками, реально выполняющими БП, а также возможность контактировать с этими людьми, при необходимости получая от них помощь и ответы на свои вопросы.

*Сочетание централизованных и децентрализованных операций.* В перепроектированных БП имеются возможности сочетать выгоды, получаемые от централизации и децентрализации в одном и том же БП. Информационная технология позволяет подразделениям действовать как полностью автономным организационным единицам, в то же время пользуясь преимуществами централизации. Например, снабжение торговых представителей на местах компьютерами класса notebook, соединенными с помощью модемов с головным отделением, дает им возможность получения немедленного доступа к хранящейся там информации. В то же время средства контроля, заложенные в программное обеспечение, используемое этими торговыми представителями при заключении контрактов, не позволяют им устанавливать неразумные цены либо задавать условия доставки (или какие-то иные условия), которые предприятие не может выполнить. Используя эту технологию, предприятия могут перепроектировать БП продаж таким образом, чтобы избавиться от бюрократического аппарата в местных отделениях, повысить независимость и уровень полномочий торговых представителей и одновременно усовершенствовать контроль за расценками и условиями продаж.

Приведенные характеристики присущи многим перепроектированным БП. Но это вовсе не доказывает, что все перепроектированные БП выглядят одинаково или что перепроектирование БП – это нечто простое и стандартное. Этого и не может быть, хотя бы потому, что некоторые характеристики конфликтуют друг с другом. На самом деле создание нового БП требует глубоких знаний старого БП, творческого подхода и дальновидности.

## **4. ОБСЛЕДОВАНИЕ И ВЕРБАЛЬНОЕ ОПИСАНИЕ СИСТЕМ**

*Для построения модели процесса необходимо либо быть его участником, либо наблюдать за ним со стороны, либо иметь о нем достаточную информацию. Мы уже построили несколько ARIS-моделей несложных подсистем и процессов, используя для этого их словесное описание. ARIS-аналитики, как правило, работают с вербальной информацией, описывающей динамическую систему и ее процессы. Эту информацию добывают зачастую они сами, например, проводя обследование моделируемого предприятия.*

*В этом разделе кратко изложены основы методики проведения обследования деятельности предприятия и подготовки словесного (вербального) описания результатов этого обследования.*

### **4.1. Обследование предприятия**

Для формального описания предприятия и его БП необходимо обследовать это предприятие и собрать достаточную информацию. При обследовании осуществляется сбор информации, которая характеризует предприятие в целом, его БП, организационно-штатную структуру, систему управления, нормативно-справочную документацию, материальные и информационные потоки и т. п.

По окончании обследования строится и согласуется с заказчиком предварительный вариант модели предприятия, детализированный до уровня основных деятельности и информационных связей между ними. В дальнейшем на основании согласованных моделей верхнего уровня осуществляется построение детальных моделей.

Необходимо отметить, что один системный аналитик должен обследовать не более двух или трех видов деятельности предприятия (таких, как учет кадров, бухгалтерский учет, маркетинг, ремонт оборудования, перевозки и т. п.) для того, чтобы тщательно в них разобраться. Современное предприятие является сложной системой, состоящей из крупных взаимоувязанных подсистем (деятельностей), а возможности человека в одновременном охвате большого количества таких подсистем ограничены, поэтому здесь в полной мере должен использоваться принцип “разделяй и властвуй”.

При проведении обследования используется следующая исходная информация: данные по оргштатной структуре предприятия, описания существующих технологий деятельности, стратегические цели и перспективы развития, результаты опроса сотрудников (от руководителей до исполнителей нижнего звена), предложения сотрудников по усовершенствованию БП предприятия, нормативно-справочная документация, данные по имеющимся на предприятии средствам и системам автоматизации.

Обследуемый элемент системы управления предприятием описывается в следующих терминах:

– *вход*, т. е. входная информация, объекты учета, структура и содержание нормативно-справочной информации, которую используют как контекстные данные для выполнения БП, первичные документы, данные или объекты, которые потребляются или изменяются в БП, ограничения, накладываемые на входную информацию, ресурсы, необходимые для выполнения БП, состав исполнителей, технических средств, оборудования, без которых невозможно реализовать БП, и т. п.;

– *выход*, т. е. выходная информация, структура и содержание выходных документов, основной результат выполнения процесса, конечный продукт процесса и т. д.;

– *функции*, т. е. правила обработки входа (например, входной информации для получения результатной информации), требования нормативных документов, которыми руководствуются при выполнении процесса производства, и т. д.

Какую же информацию нужно выявлять прежде всего во время обследования? Во-первых, должны быть выявлены все внешние объекты, с которыми моделируемое предприятие взаимодействует, технологии взаимодействия со стороны предприятия, а также информационные и материальные потоки, обеспечивающие эти взаимодействия. Во-вторых, должны быть детально выявлены реальные технологии работы предприятия. Следует помнить, что нормативно-справочная документация (если она имеется) описывает их неполно. В-третьих, должны быть определены реальные функции подразделений и их взаимосвязи и взаимозависимости, поскольку положения о подразделениях такую информацию не содержат. В-четвертых, должны быть выявлены и специфицированы все информационные хранилища (в том числе и бумажные: картотеки, архивы и т. п.). В-пятых, должна быть оценена аппаратно-техническая база предприятия, а также



исследовано работающее на ней программное обеспечение. Наконец, в-шестых, должны быть собраны статистические данные по БП предприятия.

Погрешности, связанные с недостатком и неточностью информации, усугубляются нарушением процедуры сбора информации и обработки результатов обследования. В этой связи целесообразно придерживаться следующих этапов организации и проведения обследования: формулирование целей обследования, отбор и формирование группы аналитиков (экспертов по видам деятельности), разработка модели обследования, сбор информации, анализ и обработка полученной информации, синтез недостающей информации, приведение добытой информации к форме, удобной для дальнейшего использования.

Формирование группы аналитиков направлено на разработку метода и модели сбора, анализа и обобщения информации. Сбор информации при обследовании проводится по специальной методике, в частности, путем опроса сотрудников предприятия. Этот опрос может быть организован с использованием метода анкет, метода интервью и метода групповых обсуждений.

#### **4.2. Сбор информации о предприятии**

При проведении обследования рекомендуется осуществлять *сбор документов, анкетирование, интервьюирование и групповой опрос*. Целью анкетирования, интервьюирования и группового опроса является сбор информации, которая детально характеризует БП и систему управления предприятием.

*Сбор документов* должен осуществляться на всех этапах проведения обследования. Соответствующие формы, бланки и т. п. в дальнейшем сослужат неоценимую помощь при разработке моделей предприятия. Целесообразно подготовить альбом форм с разбивкой их по деятельности предприятия.

*Метод анкет* основан на заполнении сотрудниками подразделений обследуемого предприятия специальных опросных листов – анкет. Цель анкетирования – сбор информации, которая характеризует предприятие и его систему управления как в целом, так и по отдельным элементам. *Анкеты* – это перечни тематических вопросов, ответы на которые готовит и пишет специалист (группа специалистов) обследуемого предприятия. Анкеты предназначены для того, чтобы составить общее представление о деятельности предприятия, подготовить перечни систематизированной информации о предприятии и спланировать работу группы

экспертов, обследующих предприятие.

*Анкетирование* предполагает выезд группы системных аналитиков на предприятие. Анкеты должны раздаться руководителям структурных подразделений и содержать графы для идентификации фамилии и должности анкетированного. Отдельно излагается просьба приложить шаблоны документов, с которыми работают сотрудники соответствующего подразделения. Список вопросов должен быть ограничен (не более 15-20) с тем, чтобы вся анкета не занимала более двух листов.

Необходимо иметь в виду, что излишняя детализация обследуемых объектов (факторов, проблем) может привести к снижению надежности полученной информации. Вопросник (анкета) должен обеспечить правильное, единственное толкование ответа на каждый вопрос в виде количественной оценки. При этом возможны различные варианты оценок и методов упорядоченности (доли, проценты, баллы). Анкета представляет собой структурно упорядоченный набор вопросов (по блокам, разделам), каждый из которых связан с конкретной задачей опроса.

Примерный вариант анкеты при сборе информации об офисной деятельности приведен ниже:

- *ФИО руководителя подразделения, телефон;*
- *координаты контактного лица (к кому в отсутствие или при занятости руководителя можно обращаться);*
- *основные функции подразделения;*
- *какая информация поступает из других подразделений (заявки, запросы, отчеты и т. п.);*
- *какая информация передается в другие подразделения;*
- *какая информация формируется (“рождается”) в подразделении;*
- *какими внешними предприятиями (банк, заказчик, поставщик и т. п.) взаимодействует подразделение и какой информацией обменивается;*
- *физическое представление информационных потоков и хранилищ (документ, дискета, сеть, журнал, картотека и т. п.);*
- *время хранения информации;*
- *документы от и для руководства;*
- *штатная структура и квалификация кадров*
- *техническое оснащение подразделения (компьютеры, сеть, модем и т. п.);*
- *используемые программные продукты.*

Просьба приложить:

- 1) положение о подразделении;
- 2) набор документальных форм без внутреннего наполнения, т. е. используемые формы, бланки и др.

*Интервьюирование* является важнейшим и необходимым методом сбора информации, только с его помощью возможно разобраться во всех тонкостях применяемых на предприятии технологий. Современное предприятие является сложнейшей системой, как оно функционирует в целом, не знает ни один человек. Конечно, руководство представляет ситуацию в целом, с другой стороны, клерк досконально знает свою деятельность, но полной картины не имеет никто. И только интервьюирование представителей всех звеньев организационной структуры позволит выявить и в дальнейшем формализовать эту картину.

*Метод интервью* подобен методу анкет, но ответы в опросных листах фиксирует эксперт, проводящий обследование. При этом каждый ответ может быть проверен и уточнен с помощью дополнительных вопросов. Вопросы должны дополнять друг друга, подтверждая, исправляя или опровергая ответы на предыдущие вопросы. Во время интервью очень часто можно открыть новые взаимосвязи и ограничения. В целях качественного сбора информации беседой можно управлять, применяя разные схемы интервью на основе базовых вопросов опросных листов. Все возникающие во время беседы дополнительные вопросы, не предусмотренные опросными листами, записываются в опросные листы вместе с ответами на них. Опросные листы могут быть оформлены как перечни тематических вопросов с заранее указанными вариантами ответов или без них.

*Метод групповых обсуждений* заключается в изучении деятельности предприятия на основе проведения встреч (совещаний), на которых присутствует группа определенных специалистов обследуемого предприятия. Этот метод подобен методу интервью, но в отличие от него тематические опросные листы используются как опорный материал для проведения опроса группы специалистов, т. е. сбор информации по тематическим опросным листам осуществляется путем опроса не каждого специалиста в отдельности, а группы специалистов предприятия.

### 4.3. Вербальное описание систем

Построение моделей “целевых” систем и процессов, например, моделей планируемых (задуманных) бизнес-деятельностей, удобно начинать с их словесного описания. Рассмотрим особенности такого описания.

Процессы являются фундаментальным свойством существования динамической системы, т. е. существование любой динамической системы невозможно без наличия протекающих в ней процессов. Система де-

градирует, если ее процессами никто не занимается, и распадается, если разрушаются ее процессы. Поэтому для описания экономической системы (организации) необходимо описать ее БП. При словесном (вербальном) описании достаточно описать задачи, которые решаются при выполнении БП. Грубо говоря, словесное описание системы – это описание задач, которые решаются в системе. Вербальное описание не должно содержать особенности процесса решения задачи и тем более реализации этого процесса. Излишние детали навязывают определенную реализацию и препятствуют выбору другой, возможно, более эффективной реализации.

Описать БП – значит, прежде всего, подобрать точные понятия, адекватные этому БП. Описание должно состоять из надежных, ясных и недвусмысленных понятий. Работа с понятиями – главное в построении и использовании описаний. Понятия, с помощью которых строятся описания, называются понятийными средствами описания системы. При описании от понятийных средств требуются выразительность, богатство и гибкость.

Основными чертами или свойствами описаний являются точность, ясность и полнота. Полнота описания означает, что ничто существенное из деятельности организации в ней не упущено, не забыто. Описание в принципе должно охватывать все существенные аспекты системы (свойство полноты). Точность описания называют также формальностью или однозначностью. Третье основное свойство описания – ясность, называют также понятностью или читабельностью. Оптимальной в смысле ясности считают то описание, которое самым прямым, простым, естественным и понятным способом описывает задачи. Описание “на естественном языке” обычно не обеспечивает эти требования, но с него, как правило, начинается описание системы (процессов).

В описании не должно быть ничего лишнего. Нежные детали затемняют описание. Это требование может вступать в противоречие с ясностью. Краткость – это хорошо, но не в ущерб ясности.

Фактически описание системы (процесса) нередко полностью или частично существует в головах разработчиков, а не на бумаге. Они устно обмениваются информацией, обсуждают неясности и согласовывают детали при непосредственном общении. Это хорошо, но описания надо превратить в явные и зафиксированные на том или ином носителе материалы, потому что описание системы – это очень важный документ, сопровождающий систему на всем ее “жизненном пути”. Подготовка и документирование полных и правильных описаний – самая ответственная задача, например, в раз-

работке ИС. На этапе проверки эффективности внедрения ИС описание служит для выяснения того, решает ли ИС поставленные задачи. Описание является частью “контракта”, соглашения между заказчиком ИС и ее разработчиком.

Людам свойственно ошибаться, ошибки в описаниях неизбежны. Поэтому необходимы многократные чтения или рецензирование описаний разными людьми.

Ценное человеческое качество, которое следует культивировать в описаниях, – честность. Не закрывать глаза на “дыры” в описаниях, уметь признать слабость или отсутствие обоснования выбранного решения, не замалчивать чужую идею в угоду собственной, отказаться, когда надо, от привычного понятия или шаблона, от соблазна сократить описание или выдать “темное” за ясное и т. д. – для этого нужна честность. Очистить суть дела от рекламной шелухи, выявить и использовать то, что полезно по существу, а не то, что вошло в моду или поддерживает чей-то престиж, – для этого тоже нужна честность.

И, наконец, как мощный человеческий фактор следует отметить здравый смысл. Например, в описании процесса обучения не нужно указывать, что оно не должно осуществляться в ночное время. На то и здравый смысл, который должен удерживать от крайностей чрезмерной формализации и детализации или, наоборот, недосказанности и незаконченности.

Разработка описаний представляет собой манипулирование понятиями, именами и определениями. Имена и техника именования играют в описаниях очень важную роль. Желательно писать доходчиво и объяснять убедительно. Бывает, что человек, имеющий ясные понятия, путано их излагает и, наоборот, ясно излагает путаницу, царящую в его голове.

При введении нового понятия исключительно важна роль примеров. Определения нужно давать обязательно с примерами. Примеры нередко обнаруживают несоответствие формулировки задуманной идеи. Хороший пример или набор примеров часто может заменить определение, легко извлекаемое из примеров. Бывают случаи, когда трудно дать общее точное определение, но легко объяснить суть дела на точных примерах. Общие понятия возникают как обобщения примеров.

Иногда “уловить мысль” можно с помощью вариантов понятия, отличающихся друг от друга более или менее второстепенными деталями.

Понятия определяются через базисные и друг через друга. Имея набор определений, можно установить взаимосвязи между участвующими в них понятиями, включая базисные.

Насколько элементарными, простыми или очевидными следует взять базисные понятия, что надо явно определить в описании, а что оставить без определения, – это приходится решать, исходя из опыта и здравого смысла, с учетом конкретной ситуации и ее участников. С одной стороны, нельзя требовать, чтобы всё формально определялось. То, что логически просто, нет смысла определять. Достаточно объяснить, что именно имеется в виду под данным словом. В каждой области имеется момент, когда становится бесполезным дальнейшее уточнение деталей, т. е. дополнительные подробности лишь вносят путаницу.

Надо предусмотреть, чтобы предполагаемые, неявные, подразумеваемые “по умолчанию”, не присутствующие прямо в описании понятия (а это обычно первичные понятия), одинаково понимались всеми. Пояснения элементарных и определения неэлементарных и базисных понятий должны быть где-то фиксированы и доступны.

Вводя новое понятие, ему присваивают имя-термин. Многие употребительные понятия имеют более или менее установившиеся названия-термины, нередко не одно, а несколько (синонимы). Строя систему определений, надо тщательно следить за использованием новых и базисных терминов, за синонимами, если они допускаются, и не допускать омонимии, т. е. когда один термин используется в качестве имени (названия) разных понятий.

#### **4.4. Этапы вербального описания предприятий**

На стадии вербального описания предприятия выполняются следующие работы.

1. Формулирование (уточнение) миссии предприятия.
2. Определение критических факторов успеха (5 – 7 факторов): длительность, издержки, качество продукции, качество обслуживания (сервис) и т. д.
3. Выявление основных, вспомогательных и других БП, как существующих, так и перспективных (10–15 БП).
4. Оценка БП по степени реализации критических факторов успеха.
5. Ранжирование БП с указанием приоритетов.
6. Описание отличительных особенностей БП.
7. Описание существующих обеспечивающих, производственных и информационных технологий.
8. Описание возможных сценариев развития предприятия: появление новых технологий, ресурсов, изменений поведения клиентов, партнеров, конкурентов.
9. Определение ограничений, связанных с уровнем квалификации персонала, технической оснащенности производства и т. д.

10. Определение внешних рисков, обеспечения финансовыми ресурсами, надежности партнеров.

Остановимся вкратце на описании (уточнении) миссии организации.

*Миссия организации* – это сформулированное основное социально-значимое долгосрочное функциональное назначение организации. Миссию можно сформулировать, отвечая на вопросы: "Кто мы? Что мы делаем? Какие наши цели? Куда мы стремимся?"

Например, миссию Электротехнического факультета (ЭТФ) ЮУрГУ можно сформулировать следующим образом.

*ЭТФ – центр образования и науки в городе Миассе, в котором на высоком научно-методическом уровне предоставляется опережающее университетское образование с уклоном на управление в технических и экономических системах на основе использования передовых информационных технологий, где постоянно стремятся овладевать мировыми достижениями в прикладной информатике и математике с целью их использования в учебном процессе, а также в научно-исследовательской и проектно-конструкторской работе.*

В этой формулировке есть недостатки. Во-первых, она длинная, во-вторых, дважды указано на информационную составляющую и науку. Следующая формулировка миссии ЭТФ лишена этих недостатков.

*Миссией ЭТФ, как центра образования и науки в городе Миассе, является удовлетворение потребностей предприятий и жителей города в опережающем университетском образовании на высоком научно-методическом уровне в области управления экономическими и техническими системами с использованием передовых информационных технологий.*

Но в этой формулировке, во-первых, не указано, куда стремится ЭТФ, во-вторых, не указана научная деятельность ЭТФ. Возможна следующая итерация формулировки миссии ЭТФ.

*ЭТФ – центр образования и науки в городе Миассе, в котором удовлетворяются потребности предприятий и жителей города в опережающем университетском образовании на высоком научно-методическом уровне в области управления экономическими и техническими системами с использованием передовых информационных технологий. Здесь стремятся научить студентов самостоятельно добывать новые знания и использовать их на практике, в том числе в фундаментальных и прикладных научных исследованиях и проектно-конструкторских работах.*

Выполнение миссии организации обеспечивается ее *стратегическими процессами*. Цель процесса – ожидаемый (требуемый) будущий результат, на дости-

жение которого направлен процесс.

После утверждения миссии организации ставятся ее *стратегические цели* и формируется *стратегический план*, направленный на достижение этих целей. Этот план либо оформляется в виде документа, либо мысленно представляется как единое видение и понимание руководством организации путей развития этой организации. Чем крупнее организация, тем выше вероятность, что ее стратегический план описан и задокументирован.

Целью может быть, например, выполнение каждой функции процесса с минимальными общими затратами и высоким качеством. Для этого необходимо оценить каждую функцию, т. е. рассчитать ресурсные затраты на ее исполнение и оценить качество ее исполнения.

При описании БП важно описать цели этого БП, затем описать взаимодействия с другими БП. Далее определить используемые ресурсы, разработать методику оценки (самооценки) этого БП и приступить к построению его моделей. БП системы необходимо ранжировать. Для этого рекомендуется определить несколько критериев качества (целей) и критические факторы успеха, а затем ранжировать БП по количеству тех показателей качества и целей, которые они обеспечивают.

Рассмотрим этап выделения и классификации БП. Как правило, основу для классификации БП составляют *основные БП, обеспечивающие БП, БП развития и БП управления*.

*Основными БП* являются те, которые ориентированы на производство продукции или оказание услуг, представляющих ценность для клиента и обеспечивающих получение дохода для предприятия. Эти БП производят "Выходы" процессов. Как правило, основных БП на предприятии немного, обычно не более десяти. В состав основных входят, например, следующие БП: производство основной продукции, производство товаров и изделий из отходов основного производства, продажа энергоресурсов сторонним организациям.

*Процессы управления* – это БП, которые охватывают весь комплекс функций управления на уровне каждого БП и бизнес-системы в целом, т. е. взаимосвязанного множества всех БП предприятия.

К *процессам развития*, как правило, относятся процессы совершенствования производимого продукта или услуги, технологии, оборудования, а также инновационные процессы.

*Обеспечивающие БП* – это вспомогательные процессы, которые предназначены для обеспечения выполнения основных БП. В общем виде они обеспечивают ресурсами все БП предприятия. В отличие от основных количество обеспечивающих БП достигает нескольких десятков.



Для каждого обеспечивающего БП важно определить перечень основных БП, БП развития, а также других обеспечивающих БП, технологию выполнения которых обеспечивает данный обеспечивающий БП.

К *общим обеспечивающим БП* можно отнести, например, следующие: формирование спроса на товары и услуги, планирование и экономический анализ, финансовое обеспечение, снабжение, юридическое обеспечение, обеспечение качества продукции, обеспечение транспортом, проведение ремонтно-строительных работ, обеспечение персоналом, обеспечение охраны труда, техники безопасности и гражданской обороны, социальное обслуживание персонала, охрана объектов и материальных ценностей, финансовое обеспече-

ние, оперативное планирование, информационное обеспечение, охрана окружающей среды, хозяйственное обеспечение, бухгалтерский учет и отчетность, обеспечение режима, медицинское обеспечение, дело-производство и др.

При классификации для каждого БП определяется состав его участников. Важное место при определении участников занимает владелец БП, как правило, должностное лицо (топ-менеджер), а также состав структурных элементов или должностных лиц организационной структуры предприятия, имеющих соответствующие роли.

В целом, вербальное описание можно закончить тогда, когда на его основе можно приступить к формальному описанию, например, построению ARIS-диаграмм.

## 5. АРХИТЕКТУРА ИНТЕГРИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ (АРИС)

*АРИС (АРхитектура интегрированных Информационных Систем) является концепцией и методологией, в которой интегрировано множество различных методов, от описания БП до реализации прикладных систем, автоматизирующих выполнение этих БП, а также документирования знаний по системам и их процессам. Архитектура ARIS (“дом ARIS”) – это совокупность технологий, которые обеспечивают разработку и совершенствование системы управления предприятием, а также проектирование и создание интегрированных ИС ее поддержки.*

### 5.1. АРИС-методология моделирования систем

После вербального описания системы (процессов) переходят к более строгому (формальному) описанию, т. е. к моделированию (построению моделей). Термин “моделирование” применяется весьма широко и имеет несколько значений. Мы определили моделирование как процесс построения модели. Но под моделированием также понимают и процесс исследования модели функционирования системы. Поэтому, если модели еще нет, то моделирование – это процесс построения модели, если модель уже есть, то моделирование – это процесс исследования (анализа) функционирования системы (вернее, ее модели). Часто этот процесс называют имитационным моделированием.

Для моделирования можно использовать различные методы. *Метод* моделирования включает язык и процесс моделирования. *Процесс* моделирования – это последовательность действий (операций, функций), которые необходимо выполнить для построения модели. *Язык* моделирования – это графическая нотация, используемая методом для построения модели. *Нотация* отражает синтаксис языка моделирования, т. е. содержит множество графических объектов, используемых при построении моделей (синтаксис русского языка образуют 33 буквы и различные знаки препинания). *Модель* – это построенный образ (заместитель) оригинала. Различают двухмерные (схемы) и трехмерные (макеты) модели. В АРИС рассматриваются двухмерные модели, т. к. они представляются в виде диаграмм (картинок).

Если по роду вашей деятельности достаточно только знакомиться, обсуждать, изучать или анализировать БП, то не обязательно знать процесс моделирования, но необходимо знать язык моделирования. Для формального описания БП необходимо знать предметную область (что описываем), а также язык и процесс моделирования.

АРИС предлагает методологию моделирования деятельности предприятия. Построение модели в

АРИС вынуждает системного аналитика глубоко разбираться в моделируемом объекте, связать воедино разрозненные и неполные данные о нем. АРИС играет роль “активизатора”, заставляющего аналитика искать недостающую информацию, разбираться в неточностях и противоречиях. Ведь формирование исходных данных процесса и подбор количественных характеристик функций или информационных потоков невозможно осуществить без подробных знаний о предмете.

В АРИС-моделях можно найти ответы на следующие вопросы: кто из сотрудников предприятия что делает или должен делать; каким образом это следует делать; какие ресурсы для этого необходимы; какой информацией обмениваются участники деятельности; как они согласовывают свою деятельность; какие знания необходимы для выполнения работ.

В АРИС используются различные подходы к отображению моделей БП, среди которых выделяются функциональный и объектно-ориентированный. В функциональном подходе главным структурообразующим элементом является функция (действие, операция), в объектно-ориентированном подходе – объект. Функциональный подход ориентирован на отображение последовательности функций без явного описания схемы взаимодействия объектов. Здесь еще до реализации указывается последовательность (в некоторых типах моделей и логика) выполнения операций БП, в которых трудно увидеть конкретные альтернативы построения проектируемого БП. Наоборот, объектно-ориентированный подход отражает только общую схему взаимодействия объектов без детализации последовательности выполнения функций, но с описанием условий и событий, при которых объекты инициируют выполнение конкретных функций БП.

Ключевым понятием АРИС-методологии применительно к экономике является понятие БП. При этом все сущности в АРИС равноправны, т. е. при описании БП функции, организационные единицы, входы-выходы и данные равноправны. В АРИС выделение какой-либо одной из сторон БП, например, функциональной



или объектной, и “подстройка” под нее других сторон не практикуется. На базе понятия БП строится по существу теория и технология разработки единой и целостной ИС, ориентированной на нужды бизнеса.

Получаемая в рамках АРИС модель ИС масштаба предприятия достаточно сложна. Но возможно постепенное описание, улучшение и внедрение отдельных БП, начиная с наиболее важных для предприятия. После этого появляется необходимая предпосылка для внедрения всех остальных БП. В первую очередь следует заботиться о целостности БП, а не ИС в масштабе предприятия, т. е. нет смысла покупать и внедрять ИС до анализа и оптимизации БП. Последовательное проведение в жизнь такой стратегии приводит к созданию на предприятии единой и целостной ИС, построенной на принципах АРИС.

АРИС служит корпоративным средством описания деятельности организации. С его помощью можно задокументировать все БП предприятия, выявить в них недостатки, мешающие эффективной работе предприятия, получить на основе моделей БП нормативные документы для поддержки и автоматизации БП и т. д.

Сведения, сосредоточенные в построенных моделях БП, позволяют получать любую детальную информацию для дальнейшей работы, например, должностные инструкции сотрудников, состав используемых документов, условия выполнения тех или иных действий сотрудников.

Разработав альтернативные БП и определив их качественные характеристики, можно выбрать наиболее эффективные варианты построения БП.

После моделирования и оптимизации БП появляется возможность внедрения стандартного программного обеспечения, настройки комплексной системы управления корпорациями (R/3, BAAN IV) или разработки собственного программного обеспечения. На основе построенных моделей можно осуществить функционально-стоимостной анализ, внедрить систему workflow или подготовиться к сертификации по стандартам ISO 9000 и выше.

АРИС отвечает современным требованиям, позволяя моделировать, анализировать и оптимизировать БП. Это хорошее средство для эффективной разработки систем управления качеством, ориентированных на БП. Системы управления качеством включают организационную структуру, субъекты ответственности, процедуры, процессы и средства, необходимые для управления качеством. В стандарте системы управления качеством (TQM) сформулированы критерии для определения качества БП. Компания может удостовериться в степени своего соответствия этим критериям, полу-

чив надлежащий сертификат, который свидетельствует о качестве БП. Для моделей TQM центром приложения критериев оценки являются БП. В ARIS можно документировать каждый базовый элемент TQM, фигурирующий в стандарте ISO 900X. Сюда входят описание обязанностей в рамках организации (должностные инструкции), идентификация продукции, ее приобретение, изготовление и сопровождение, управление документооборотом, а также перемещение, хранение, упаковка, отправка продукции и многое другое.

Используя АРИС, можно описать процесс моделирования, т. е. унифицированную последовательность действий при построении моделей. Процесс объектно-ориентированного моделирования уже разработан, описан и получил название Objectory. Для применения АРИС или других методологий не обязательно, но желательно использовать Objectory.

Для создания системы сопровождения управления предприятием с использованием АРИС-технологий осуществляется следующее: классификация БП, создание полных и согласованных моделей БП и их автоматизированная поддержка в актуальном состоянии, автоматическое документирование БП, мониторинг выполнения БП и анализ отклонений, автоматизированная поддержка создания и сопровождения документации по системе управления качеством, автоматизированное предоставление информации для проведения аудита, использование единой ИС для выполнения проектов по реинжинирингу БП, проектированию ИС, функционально-стоимостному анализу, имитационному моделированию и др., автоматизированная поддержка процесса перехода от менеджмента качества к глобальному управлению качеством, создание и управление корпоративными знаниями и опытом.

## 5.2. Некоторые особенности АРИС-моделирования

Моделирование – процесс творческий, и поэтому он не поддается жесткой регламентации. Однако, соблюдение некоторых принципов позволяет повысить эффективность использования разрабатываемых моделей. Поэтому, используя АРИС, рекомендуется придерживаться следующих принципов моделирования.

*Принцип корректности.* Корректность модели зависит от правильности ее семантики и синтаксиса. ARIS располагает широким набором правил для проверки корректности моделей. Семантическая полнота и корректность модели определяются тем, насколько адекватно она отвечает структуре и поведению моде-

лируемой системы. Соответствие этим требованиям можно подтвердить после проведения имитационных экспериментов с полученными моделями.

*Принцип релевантности.* Следует моделировать лишь ту статику и динамику реальной системы, которые соответствуют назначению модели. Модель не должна содержать информации больше, чем необходимо для достижения поставленных целей моделирования.

*Принцип соизмеримости затрат и выгод.* К числу факторов, определяющих эффективность моделирования, относятся, во-первых, объем затрат, необходимых для создания моделей, во-вторых, полезность моделей, в-третьих, продолжительность их использования. Необходимо стремиться к максимизации полезности и продолжительности использования моделей, но минимизации затрат на их создание.

*Принцип прозрачности.* “Прозрачность” гарантирует понятность и удобство модели для пользователей. АРИС-модели разбиваются на виды, типы и уровни представлений, что облегчает понимание конкретных аспектов моделирования.

*Принцип сравнимости.* Модели, созданные на базе согласованной концептуальной инфраструктуры и единого языка моделирования, сравнимы между собой, если, конечно, имена объектов отвечают установленным соглашениям и если использовались идентичные объекты моделирования, а также эквивалентные степени детализации.

*Принцип систематизированной структуры.* Этот принцип в качестве обязательного условия предполагает возможность интеграции моделей разных типов. Для этого требуется единая метамодель, объединяющая различные типы представлений, что и обеспечивает информационная модель ARIS.

Различают модели БП “как есть”, т. е. модели существующих БП, и “как должно быть”, т. е. целевые модели БП. Разработка целевых моделей БП заканчивается построением реальной и идеальной модели БП. Идеальная модель может быть достигнута в перспективе, и к ней следует стремиться. Реальная модель может быть достигнута в обозримом будущем с учетом имеющихся ресурсов. Причем, реальная модель БП должна быть такой, чтобы можно было в перспективе перейти к идеальной модели. Таким образом, на основе моделирования БП выбираются наиболее эффективные варианты их организации с точки зрения реализации ключевых факторов успеха.

При построении моделей не рекомендуется использовать сразу все доступные синтаксические и семантические средства. Нужно начинать с самого про-

стого. Например, в диаграммах классов для начала достаточно использовать обобщения, ассоциации, атрибуты и операции, а затем более сложные средства (множественную и динамическую классификацию, дискриминаторы, агрегацию, композицию, производные ассоциации и атрибуты и др.).

Для определения вариантов использования модели необходимо перечислить все внешние (запускающие) события, на которые моделируемая система должна реагировать. При этом желательно выделить события, на которые система реагирует автоматически (без вмешательства человека) и, наоборот, вызывает чисто пользовательскую реакцию.

Надо стремиться строить повторно используемую модель системы, обеспечивающую последующие расширения, где новые варианты использования могут постепенно добавляться и включаться в модель, обеспечивая функциональное расширение модели и ее информационное наполнение. Не следует даже пытаться строить исчерпывающую модель всей системы “одним махом”. Рекомендуется взглянуть с разных точек зрения на предметную область (систему, процессы) и построить различные (возможно, плохо связанные) диаграммы. Затем уже думать о взаимосвязях между ними и их согласовании.

Необходимо начинать с моделей высокого уровня, в которых отсутствуют детали. Учет множества деталей может привести к “аналитическому” параличу.

Команда, которая занимается моделированием БП, может состоять из двух – четырех человек. В эту команду должен входить минимум один разработчик (знаток методологии) и один эксперт (специалист) моделируемой экономической подсистемы. Анализ БП становится осмысленным только при участии знатока этого БП, например, ответственного исполнителя, хозяина БП и т. п.

Построенные модели БП используются для их анализа, усовершенствования, оптимизации, автоматизации и многого другого. При усовершенствовании организационной структуры системы осуществляется разработка должностных инструкций, обучение персонала, подготовка рабочей документации и т. д. При создании новой ИС осуществляется разработка и наполнение БД, установка системы телекоммуникации, настройка и отладка программных модулей и т. д.

Для динамического анализа БП используются методы имитационного моделирования, которые позволяют генерировать конкретные бизнес-случаи выполнения БП на заданном интервале времени. При этом большое значение придается анализу “узких мест” в организации БП, связанных с перегрузкой ресурсов, образованием очередей или, наоборот, недогрузкой ресурсов.

Модели БП используются для анализа их реализации с точки зрения достаточности ресурсов и эффективности их использования. Для этого применяются методы стоимостного анализа функций и динамического имитационного моделирования. Стоимостный анализ функций позволяет выявить, во-первых, наиболее трудоемкие и затратные функции, во-вторых, функции, не вносящие вклад в образование прибыли, в-третьих, функции с низким коэффициентом использования ресурсов.

В АРИС-методологии заложена возможность использования моделей БП для сборки прикладных программных систем, поддерживающих инфраструктуру и управление бизнесом. Для этого используется идея компонентного программного обеспечения, которая заключается в сборке прикладных систем из отдельных стандартных компонентов, разработанных разными поставщиками. Концепция использования компонентов тесно связана с основными принципами объектно-ориентированного подхода.

Объекты АРИС объединяются в более крупные единицы, так называемые “бизнес-объекты”, включающие прикладную информацию о взаимодействии внутренних объектов. Бизнес-объекты включают БП вместе с соответствующими данными и функциями, выполняемыми для их обработки. Таким образом, бизнес-объекты содержат несколько объектов, описывающих бизнес.

Бизнес-объекты группируются в пакеты, так называемые “бизнес-компоненты”. На первом этапе вся система R/3 включала лишь три компонента (человеческие ресурсы, логистику и финансы). Позднее было добавлено еще с десяток вновь разработанных компонентов.

На рис.5.2.1 иллюстрируется объектно-ориентированное описание БП “обработка заказа”. Здесь представлены объекты, их имена, атрибуты и методы. Показан также обмен сообщениями, т.е. показаны передаваемые данные и данные, содержащиеся в ответе.

Появление компонентного программного обеспечения открывает новые возможности для разработки программных систем информационной поддержки и управления бизнесом. В производстве аппаратного обеспечения давно используется горизонтальный принцип, при котором процессоры, периферийные устройства, операционные системы и т. д. изготавливаются отдельно, а затем компонуются в систему. Аналогичные технологии развиваются и в области разработки прикладного программного обеспечения. Применительно к аппаратным средствам предпосылкой такого развития стало создание стандартов на процессоры, операционные системы, БД и коммуникационные сети. Поэтому для горизонтальной разработки прикладных систем создаются

стандарты на компоненты и инфраструктуры. Такие стандарты позволят адаптировать компоненты, выпускаемые разными производителями, к различным БП.

Отбор и замена компонентов производится по принципу “лучших представителей своего класса”, а при сборке центр внимания сосредоточивается на ноу-хау самого бизнеса и возможности его моделирования в соответствии с методом сборки. Главное условие для правильной сборки – наличие исчерпывающего документального описания компонентов.

Важнейшими условиями развития компонентного программного обеспечения являются создание эффективных механизмов защиты и определенная культура отношений, основанная на принципах доверия и сотрудничества между поставщиками компонентов и поставщиками готовых решений.

В заключение отметим, что использование инструментов АРИС-моделирования обеспечивает соблюдение стандартного требования, согласно которому все модели БП хранятся в едином репозитории и в любое время находятся в распоряжении всех сотрудников предприятия. Источниками информации для таких хранилищ служат различные проекты, связанные с использованием БП. Это могут быть проекты по реинжинирингу, сертификации, пооперационному исчислению стоимости, внедрению стандартного программного обеспечения, компонентной сборке прикладного программного обеспечения и т. д.

### 5.3. Этапы описания системы

Увеличение масштаба экономической системы приводит не только к ее количественному, но и к качественному усложнению. Чем больше участников в системе, тем сложнее и многообразнее функциональные, организационные и информационные взаимосвязи между ними. Практический опыт показывает, что чем больше масштаб создаваемой или модернизируемой системы, тем больше в ней удельный вес “постановки” целевых задач и меньше удельный вес их “реализации”. Именно поэтому, начиная с некоторого масштаба экономической системы, формализация процесса постановки задач становится неизбежной. В экономических системах вопросы формализации постановки рассматриваются в теории БП.

Достижения в области коммуникационных технологий и программного обеспечения привели к появлению на предприятиях распределенной вычислительной среды, объединяющей в своем составе компьютеры всех классов, от персональных до мэйнфреймов (больших ЭВМ), в которых аппаратные платформы и программное обеспече-

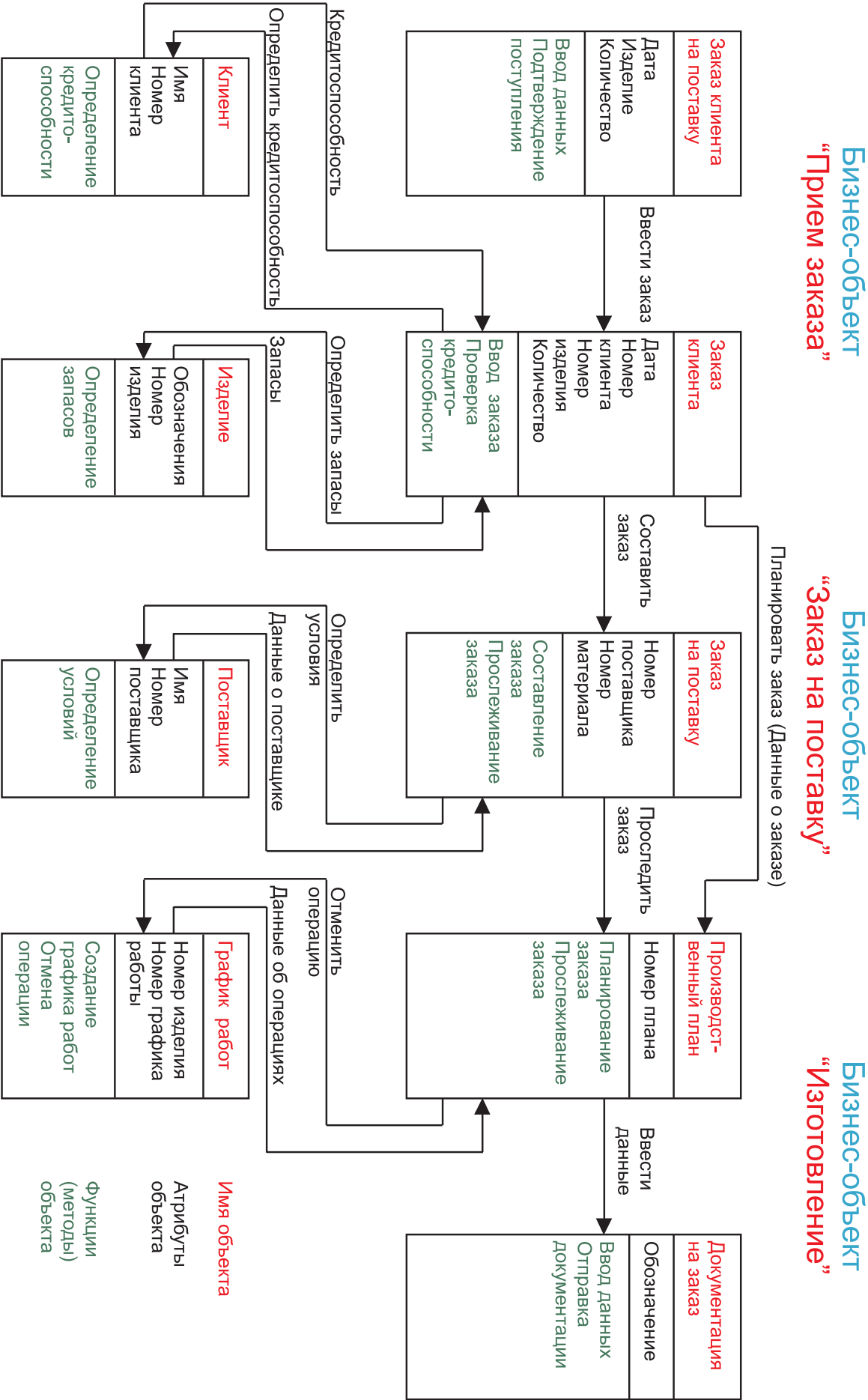


Рис.5.2.1 Объектно-ориентированное представление БП “Обработка заказа”



ние могут существенно отличаться друг от друга. Сочетание единых централизованных БД, типичных для больших ЭВМ, и технологий управления, характерных для персональных компьютеров, наряду с невиданным объемом электронных коммуникаций, создало беспрецедентно сложные проблемы постановки.

С появлением средств общего назначения, прежде всего мощных СУБД, электронных таблиц, текстовых процессоров, CASE-технологий, большие системы стали проектироваться из стандартных блоков, что упростило написание приложения, но усложнило постановку.

Поэтому информатизация и автоматизация бизнес-деятельности в АРИС основана на строгом разделении этапов процесса разработки ИС по степени их приближения к информационным технологиям (ИТ). На этом пути выделяют, во-первых, этап формулирования требований (концептуальное описание системы), во-вторых, этап описания технических условий (этап спецификации), в-третьих, этап описания реализации. Это позволяет отделить семантическую (содержательную) модель эффективной бизнес-деятельности от ее технического воплощения. Семантическая модель гораздо консервативнее ИТ, поэтому будучи раз сформулированной, она может лежать в основе нескольких следующих друг за другом реализаций на базе разных ИТ. Заметим, что ИТ обновляются ежегодно и даже чаще.

Формулирование требований призвано описать прикладную задачу бизнеса, нуждающуюся в компьютеризации, на таком формализованном языке, который может использоваться как отправная точка для последовательного перевода в ИТ. Формулирование требований связано с описанием задач бизнес-деятельности и представляет собой детализированный по определенным правилам вариант формализованного описания без привязки к реализации средствами ИТ, т. е. оставляет максимальную свободу выбора ИТ на этапе реализации.

На этапе описания технических условий понятийное обрамление сформулированных требований переводится в стандартные категории (интерфейсы) систем обработки данных. Вместо самих функций определяются модули или пользовательские транзакции, которые исполняют функции. Это этап приспособления сформулированных требований к единым интерфейсам ИТ.

*Интерфейс* – это в общем случае стандарт взаимодействия. Стандарты взаимодействия человека и компьютера называются *пользовательским интерфейсом*. Стандарты общения компьютеров при передаче данных по коммуникационным каналам – *интерфейсом телекоммуникаций*. Стандарты обращения к БД – *интерфейсом запросов* или *SQL-интерфейсом*,

по имени основного используемого при этом языка.

Стандарты взаимодействия компонентов ИС также называются интерфейсами. Таким образом, на уровне технических условий происходит наложение формулировок требований на определенный набор стандартов в области обработки данных. Связь между формулированием требований и техническими условиями должна быть по существу однонаправленной. Изменение формулировок требований должно вести к изменению технических условий. С другой стороны, требования должны быть сформулированы так, чтобы изменения ИТ и связанные с ними изменения технических условий не приводили к пересмотру самих требований.

На этапе описания реализации технические условия преобразуются в конкретные элементы аппаратного и программного обеспечения, устанавливая тем самым предметную связь с ИТ. Здесь связь также в основном однонаправленная, хотя изменение характеристик конкретных ИТ может привести к пересмотру формальных или фактических стандартов, а, следовательно, и интерфейсов ИТ.

Перечисленные три этапа характеризуются различными циклами обновления. Наибольшая частота обновления у этапа реализации, т. к. ИТ меняются очень часто, и наименьшая – у сформулированных требований. Действительно, описание реализации очень тесно связано с развитием ИТ и поэтому подвержено постоянному пересмотру в результате технологических изменений, таких, как развитие новых систем БД, сетевых технологий, аппаратного обеспечения и т. д.

Этап формулирования требований имеет максимальное практическое значение, поскольку является долгосрочным вместилищем коллективного знания и опыта бизнес-деятельности. Он в то же время является отправной точкой для получения описания реализации. По этой причине акцент делается на этап разработки формулировок требований или смысловых моделей.

Уровень определения требований особенно важен, потому что он является архивом (репозиторием) бизнес-приложения и основой для разработки описания реализации. Уровень определения требований обладает самым длинным жизненным циклом, так как описывает бизнес-проблему и ее особенности, поэтому этот уровень представляет самую большую пользу для ИС. По этой причине разработка требований (семантических моделей) имеет самый высокий приоритет. Концептуальная модель описывает требования к системе и практически никак не связана с будущими ИТ.

Отметим, что необходимость системного проектирования ИС доказана опытом работы многих компаний, занимающихся разработкой и внедрением таких систем.

Основной причиной брака при разработке ИС является недостаточное внимание к начальным этапам жизненного цикла сложных систем, а именно, анализу требований и проектированию. Следует помнить, что устранение ошибки, выявленной на этапе анализа требований и проектирования, в 10 раз дешевле, чем на этапе разработки, в 100 раз – чем на этапе тестирования и в 1000 раз дешевле, чем ее устранение на этапе внедрения.

Развитие средств отображения формулировок требований имеет богатую историю. Начало ей было положено в конце 60-х гг. прошлого столетия. В этот период был выдвинут революционный тезис о том, что основой ИС является не приложение, а БД. Тем самым был поставлен на повестку дня вопрос о проектировании структуры БД отдельно от проектирования приложения как такового. Прогрессирующая сложность структур данных потребовала выработки простого и мощного языка их описания. Это и было сделано Коддом, создавшим реляционную модель данных, и Ченом, создавшим модель “сущность-отношение” для представления произвольной модели данных.

Настоящий бум средств отображения пришелся на 80-е годы прошлого столетия, что связано с бурным развитием CASE-технологий. Например, в рамках АРИС-методологии средства моделирования и отображения ИС доведены до уровня инженерной дисциплины и объединены общей методологией, охватывающей весь процесс проектирования ИС. По этой методологии описания БП поэтапно трансформируются в объекты ИТ и коммуникационных технологий. Для этого используется пятифазная модель, описывающая реализацию бизнеса посредством компьютерных систем. Перечислим эти фазы.

Фаза 1 посвящена описанию стратегических целей с ориентацией на использование ИТ. Стратегические установки включают корпоративные цели, критические факторы успеха, риски, критерии распределения ресурсов и др.

Фаза 2 посвящена описанию требуемых (целевых) моделей БП, где указываются общие объекты ИТ, например, типы БД, программных систем, коммуникаций и т. д.

Фаза 3 связана с описанием спецификации проекта, где бизнес-модели адаптируются к требованиям интерфейсов инструментальных средств реализации (программных систем, БД, сетевых архитектур, языков программирования и т. д.). На этом этапе реальные (коммерческие) программные продукты не конкретизируются.

Фаза 4 предполагает описание реализации, где разработанные требования реализуются в виде физических структур данных, аппаратных компонентов и реальных программных продуктов.

Фаза 5 связана с сопровождением системы.

После создания общего концептуального проекта бизнес-деятельности все ее БП разбиваются на модели различных типов и документируются, проходя путь от этапа определения требований к этапу описания реализации. При этом первые четыре фазы описывают создание целевых моделей бизнес-деятельности и поэтому называются “конструктивным временем”. Когда законченная система принимает работоспособный вид, наступает 5-я фаза, которая называется “реальным временем” и связана с эксплуатацией и техническим обслуживанием средств ИТ в бизнес-деятельности.

### 5.4. Виды и типы моделей

Первая стадия формализации в описании сложных систем и процессов состоит в разделении этого описания на отдельные виды. Основная цель такого разделения – уменьшение сложности описания. В одном виде описания рассматривают тесно связанные составляющие и абстрагируются от остальных, не столь тесно связанных компонентов. В АРИС-методологии выделяются пять видов описания – *организационное описание*, *функциональное описание*, *описание данных*, *описание выходов* и *процессное описание (или описание управления)*. Структура взаимосвязи организационных единиц, взаимоотношения субъектов ответственности и многое другое рассматривается в организационном виде описания. Определения функций, перечни подфункций, описание их связей и многое другое рассматриваются в функциональном виде. В описании данных рассматриваются события, информация, массивы показателей и условий, а также многое другое. В описании выходов рассматривают ресурсы, материальные и финансовые потоки и др. В процессном описании восстанавливаются связи между вышеперечисленными описаниями и воссоздается система в целом. Последний вид часто называют описанием управления (от слова control).

Разбиение описаний на отдельные виды (взгляды) позволяет не обращать внимание на многочисленные зависимости и взаимосвязи между этими взглядами. Связи между первыми видами описания включаются в общий процессный вид без избыточности и повторяемости. В результате такой *интеграции* получается комплексная модель, которая для уменьшения сложности разделена на индивидуальные описания в отдельных видах.

В теории систем различают структуру системы и ее поведение. Структура описывает статику системы, а поведение описывает ее динамику. Четыре первых вида (организационный, функциональный, выходов и



данных) описывают структуру системы. В процессном виде устанавливаются связи перечисленных видов и описывается динамическое поведение системы. Действительно, в моделях БП динамика управляется событиями и сообщениями, которые описаны в модели данных. Сами БП состоят из последовательности функций, которые описываются в функциональном виде. Исполняют функции организационные единицы, которые описываются в организационном виде моделей. Входы и выходы для каждой функции и БП в целом описываются в моделях выходного вида.

На рис.5.4.1 перечисленные виды (взгляды) описания сгруппированы в пять частей, образующих “здание” АРИС. Рассмотренные пять видов содержат описания в категориях ИТ. Последние включаются в здание АРИС в зависимости от степени использования элементов ИТ в рамках каждого вида модели. Здание АРИС иллюстрирует архитектуру ИС, состоящую из наборов моделей пяти различных видов и разделенную на описание требований, спецификацию и описание реализации, что по разному связано с категориями ИТ (от слабой связи до сильной).

Для каждого вида описания можно построить от одной до нескольких типов моделей. Классификация моделей на виды и типы служит для структурирования моделей и позволяет устранить избыточность при многократном использовании объектов в моделях.

Следует заметить, что в программной среде ARIS 5.0 реализованы только четыре вида описаний (моделей): функциональное (function view), организационное (organization view), данных (data view), управления (control view). Модели выходов пока не выделены в отдельный вид. Рассмотрим каждый из четырех отдельных видов моделей подробнее.

*Функциональное представление* содержит описание целей, выполняемых функций, перечень отдельных подфункций, а также общие взаимосвязи и связи подчиненности, которые существуют между функциями, целями и факторами успеха. *Функциональные модели* объединяют объекты, преобразующие вход в выход. Названия “функция”, “процесс” и “операция” употребляются как синонимы. Функции направлены на достижение определенных целей, поэтому модель целей также относят к функциональному виду. В прикладных системах описываются правила компьютерной обработки функции, поэтому они также относятся к функциональной модели.

В функциональном виде можно построить дерево целей. Рассмотрим один из способов классификации целей.

По масштабам различают *всеобщие, общие и частные цели*. По временному аспекту цели бывают *актуальные, достижимые в ближайшем будущем и*

*перспективные*. По содержанию различают *экономические, организационные, экологические, правовые, культурные* и другие цели. По региональному наполнению выделяют цели *мировые, на уровне страны, региональные и местные*. В зависимости от конкретных условий те или иные цели могут выдвигаться как *первоочередные*. Существуют и другие принципы и аспекты структуризации целей.

*Организационное представление* описывает взаимодействие пользователей и организационных единиц (ОЕ), а также их связи и имеющие к ним отношение структуры. *Организационные модели* служат для описания иерархической структуры организации, где группируются субъекты ответственности, ОЕ, должностей и средств, выполняющих работу над объектами, а также для многого другого.

Описание существующей системы начинается с рассмотрения ее структуры. *Структура* отражает наиболее существенные, устойчивые связи между объектами системы и их группами, которые обеспечивают основные свойства системы. Иначе говоря, структура – это форма организации системы, скелет, костяк существования системы. Вместе с тем структура системы может претерпевать определенные изменения в зависимости от различных факторов (причин).

Особенностью ОЕ обязательно является наличие физических лиц, например, операторов, регулировщиков, исполнителей, специалистов, руководителей. Таким образом, персональный компьютер не является ОЕ, но тот же компьютер с оператором может быть элементом ОЕ, если он, например, включен в диспетчерскую сеть управления. Поэтому часто ОЕ определяют как подсистему, назначением которой является согласование действий целеустремленных частей (социальных групп и личностей), а также нецелеустремленных средств и предметов деятельности, ориентированных на достижение конкретных целей, например, получения некоторого конечного продукта. В качестве ОЕ можно рассматривать не только традиционные структуры (предприятия, организации, учреждения), но и собранные для осуществления определенной кампании различные общественные фонды, политические партии, движения и т. д.

Для многих предприятий и организаций, особенно с непрерывным циклом производства, характерным элементом ОЕ являются автоматизированные системы управления, имеющие иногда весьма сложный иерархический и интегрированный характер.

Множество связанных ОЕ образуют организационную систему (ОС). Так как ОЕ отличаются крайним разнообразием, то для их исследования необходимо ис-

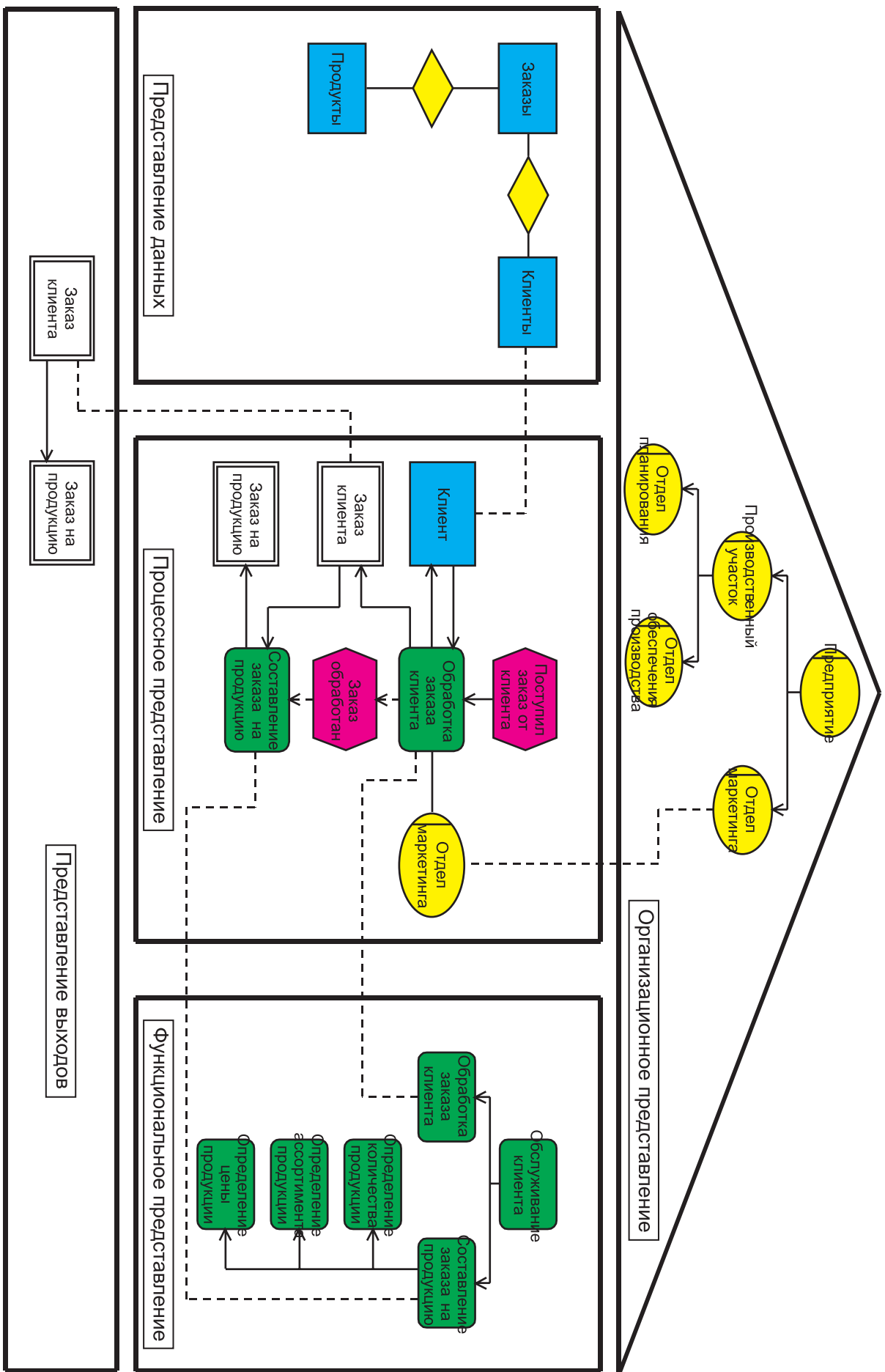


Рис. 5.4.1. Задание АРИС. Виды описания.

пользовать классификацию – выделение ОС в различные типы, классы по одному или нескольким характерным классификационным признакам.

Например, ОС могут быть разделены на *территориальные* и *отраслевые*; естественно, возможны и комбинированные – территориально-отраслевые. Другой пример классификации – *функциональные* и *целевые*. Функциональные ОС предполагают основную направленность на функционирование систем, а целевые направлены на достижение определенных целей. ОС могут быть классифицированы по *характеру разрешаемых проблем* на экономические, социальные, политические, научные, производственные и т. д. Типичными признаками политической системы являются государство, политические партии, движения и т. д. К важным классификационным признакам могут быть отнесены характеристики *социального заказчика*, потребности которого должна удовлетворять данная ОС. Социальными заказчиками могут быть международные организации, государства, коллективы предприятий, население микрорайонов и т. д.

В *моделях данных* описывают информационную среду предприятия (среду обработки данных), события, сообщения и т. д., где неявно фиксируются также объекты в виде информационных услуг.

В любой науке первичные “кирпичики” знаний впоследствии укладываются в некоторую общую систему знаний, понятий, представлений. Так, например, основа экономических знаний формируется из таких понятий, как труд, потребность, ресурсы, товар, предложение, рынок, стоимость БП и т.д. Только целенаправленный синтез знаний превращает сумму этих понятий в систему и создает реальные основы целесообразной и целенаправленной деятельности в сфере экономики. Самым сложным в сфере синтеза является выявление и исследование системы связей между сущностями, т. е. реальными процессами, явлениями и вещами, а также между соответствующими понятиями. Именно в моделях данных осуществляется формальное описание всех сущностей моделируемой предметной области и отношений между ними.

*Модели процессов (управления)* введены для описания связей между тремя предшествующими видами моделей. Интеграция этих связей в пределах отдельного представления делает возможным учесть все связи без избыточности. *Модели процессов* описывают отношения между отдельными моделями в рамках всего БП. Это позволяет отслеживать все двусторонние и многосторонние отношения между моделями различных видов, а также полностью описать БП.

Рассматривая любую динамическую систему, можно выделить два ее аспекта: функционирование и развитие.

Функционирование можно рассматривать как некоторый циклический процесс воспроизводства основных параметров экономических систем и условий их существования. При этом условия и параметры жизнедеятельности остаются более или менее постоянными. Например, функционирование органов управления, систем городского хозяйства, социальных институтов общества. Здесь имеет значение более строгая и точная повторяемость процессов, для чего необходимо стабилизировать условия их протекания. Например, функционирование жилищно-коммунального хозяйства требует проведения текущего и капитального ремонта, замены физически и морально устаревшего оборудования, дополнительных затрат на ресурсное обеспечение. Можно считать, что функционирование в известной степени соответствует простому типу воспроизводства, при котором объемы производства и потребления остаются неизменными.

Значительно большие сложности возникают при описании (моделировании) проблем развития. Развитие, естественно, не сводится к простому движению или количественному росту. Развитие означает, прежде всего, глубокие качественные изменения, которые и обуславливают переход к более высокому уровню организованности и упорядоченности системы. Оно означает необходимость включения новых факторов, новых агентов воспроизводственного цикла в процессы жизнедеятельности системы. Вместе с тем эти факторы и агенты не могут включаться в процессы жизнедеятельности иначе, чем через функционирование системы.

Понятие “*развитие*” характеризует совершенствование структуры и функций системы под влиянием, главным образом, внутренних факторов, в связи с чем поведение системы приобретает более упорядоченный и предсказуемый характер. Свойством развития обладают многие системы.

Процесс функционирования экономической системы на самом деле подвержен процессам развития и поэтому не может сохранять постоянными свои параметры и траекторию движения.

Реально процесс приобретает спиральную форму развития. На первом витке спирали в процесс функционирования включается, например, первое нововведение, которое отклоняет траекторию цикла от замкнутой окружности. В качестве такого нововведения могут быть, например, технические или организационные средства, повышающие производительность труда (компьютерная техника). На следующем витке функционирования включа-

ются другие нововведения, например, оптоволоконные линии связи, которые способствуют созданию компьютерных сетей и т. д. Вследствие этого реальный процесс жизнедеятельности системы органично выступает как единый процесс, развивающийся по спирали. В соответствии с характером данного процесса должна развиваться система и механизм управления, причем с определенным прогностическим упреждением.

Во всех видах описания и большинстве типов АРИС-моделей имеется возможность осуществлять декомпозицию, т. е. представлять модели и их объекты в виде иерархии. Это позволяет моделировать сложные системы.

*Сложные системы (СС)* – это такие системы, которые могут быть представлены совокупностью подсистем постоянно уменьшающегося уровня сложности вплоть до элементарных подсистем, выполняющих в рамках данной СС базовые элементарные функции. Процесс представления СС в виде иерархии подсистем называется *декомпозицией*. Примером декомпозиции СС является представление национальной экономики страны как совокупности подсистем – отраслей, объединений, предприятий, цехов, участков, рабочих мест. Базовой подсистемой является отдельный работник на рабочем месте. Помимо этого национальная экономика может быть представлена совокупностью региональных экономик, регионов с различным уровнем развития, с пограничными зонами, с различным уровнем дохода на душу населения и т. д., т. е. существует бесчисленное множество вариантов декомпозиции СС в зависимости от целей анализа и характера решаемых задач. Сложной системой может быть атом водорода, семья из трех человек, коллектив небольшой фирмы и гигантского концерна – все зависит от поставленных целей анализа и решаемых задач.

Как правило, известны базовые элементарные подсистемы, которые выполняют фундаментальные функции в СС. Элементарной системой организма является биологическая клетка, элементарной системой экономики – отдельный работник. Элементарная система не подлежит дальнейшему расчленению, хотя это и принципиально возможно, но уже не в рамках данной СС.

Для полноценного использования АРИС-моделей необходимо задать свойства и атрибуты моделей и составляющих их объектов, а это достаточно сложная задача, зависящая от целей использования АРИС-моделей. Свойства описывают “видимую” часть сущностей (моделей, объектов и др.). Атрибуты описывают “невидимую” часть сущностей. Здесь использовано слово “невидимая” достаточно условно, так как, во-первых, пользователь программной системы ARIS 5.0 всегда может ви-

деть и редактировать любой атрибут, во-вторых, многие из атрибутов можно показать рядом с моделью. Макросы, мастера и приложения работают как со свойствами, так и с атрибутами сущностей. В АРИС-методологии под сущностями понимается все, что достойно рассмотрения, сбора информации и описания.

## 5.5. Уровни моделирования

Модели БП можно создавать на разных уровнях абстрагирования. На рис.5.5.1 изображена диаграмма не столько реального БП, сколько некоторого обобщенного БП, представляющего собой абстракцию реально осуществляемого БП “Голосование вопроса”. На рис.5.5.2 изображен функциональный поток БП “Голосование вопроса” “Кто за проведение семинара?” на заседании кафедры СУиММ”. Это экземпляр БП на рис.5.5.1, т. к. здесь конкретизированы его участники и сформулирован голосуемый вопрос и указана конкретная кафедра.

Конкретные БП называются экземплярами. Между типом БП (рис.5.5.1) и экземпляром этого БП (рис.5.5.2) существует отношение класс-экземпляр. В экземплярах БП каждый объект конкретизируется присвоением ему фиксированного имени. Такие модели БП используются для их управления в реальном масштабе времени, например, с помощью системы workflow.

Совокупность экземпляров БП “Голосование вопроса”, где сформулирован конкретный вопрос и указаны конкретные должности (участники БП), составляет класс или тип, называемый БП “Голосование вопроса”. Конкретные БП являются экземплярами (элементами) этого класса. Класс “перенимает” основные характеристики своих экземпляров, абстрагируясь от их особенностей, в рассматриваемом примере – от формулировки вопроса, от должностей участников и организации.

На рис.5.5.1 изображен функциональный поток типового процесса “Голосования вопроса” на собрании, т. к. не конкретизированы участники собрания, их организация (цех, кафедра, факультет и т. д.) и не сформулирован голосуемый вопрос. Диаграмма на рис.5.5.2 является частным случаем диаграммы на рис.5.5.1. На рис.5.5.3 изображена модель собраний, на которых может иметь место БП “Голосование вопроса”. Конкретизируя организацию, формулировку голосуемого вопроса и должности ведущего, секретаря и участников, мы из типового процесса получим его конкретный экземпляр.

Каждый класс характеризуется именем и перечнем атрибутов, описывающих соответствующий экземпляр. Для дальнейшей характеристики классов можно перечислить прилагаемые к ним функции.

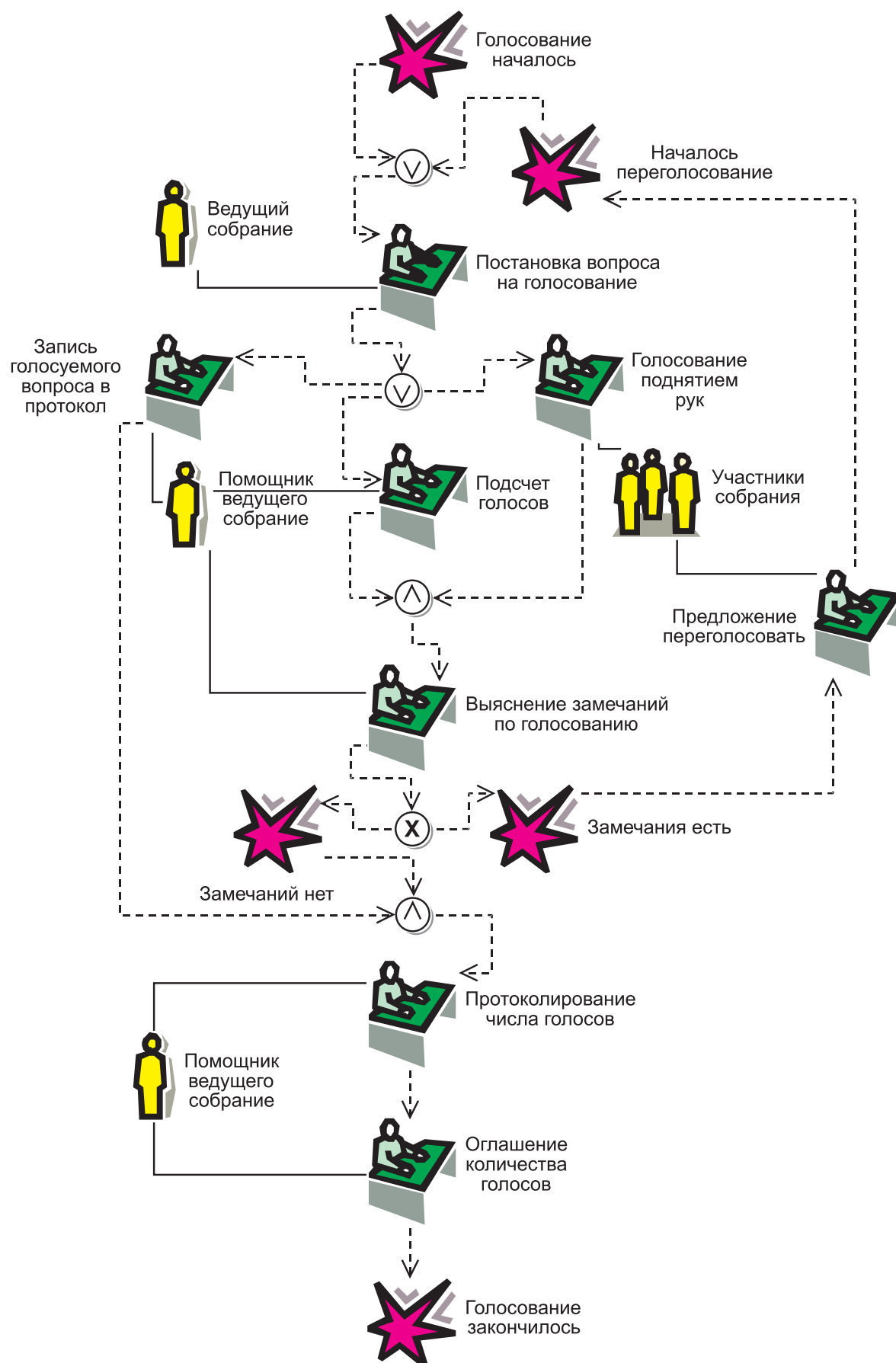


Рис.5.5.1 Офисный типовой процесс “Голосование вопроса”

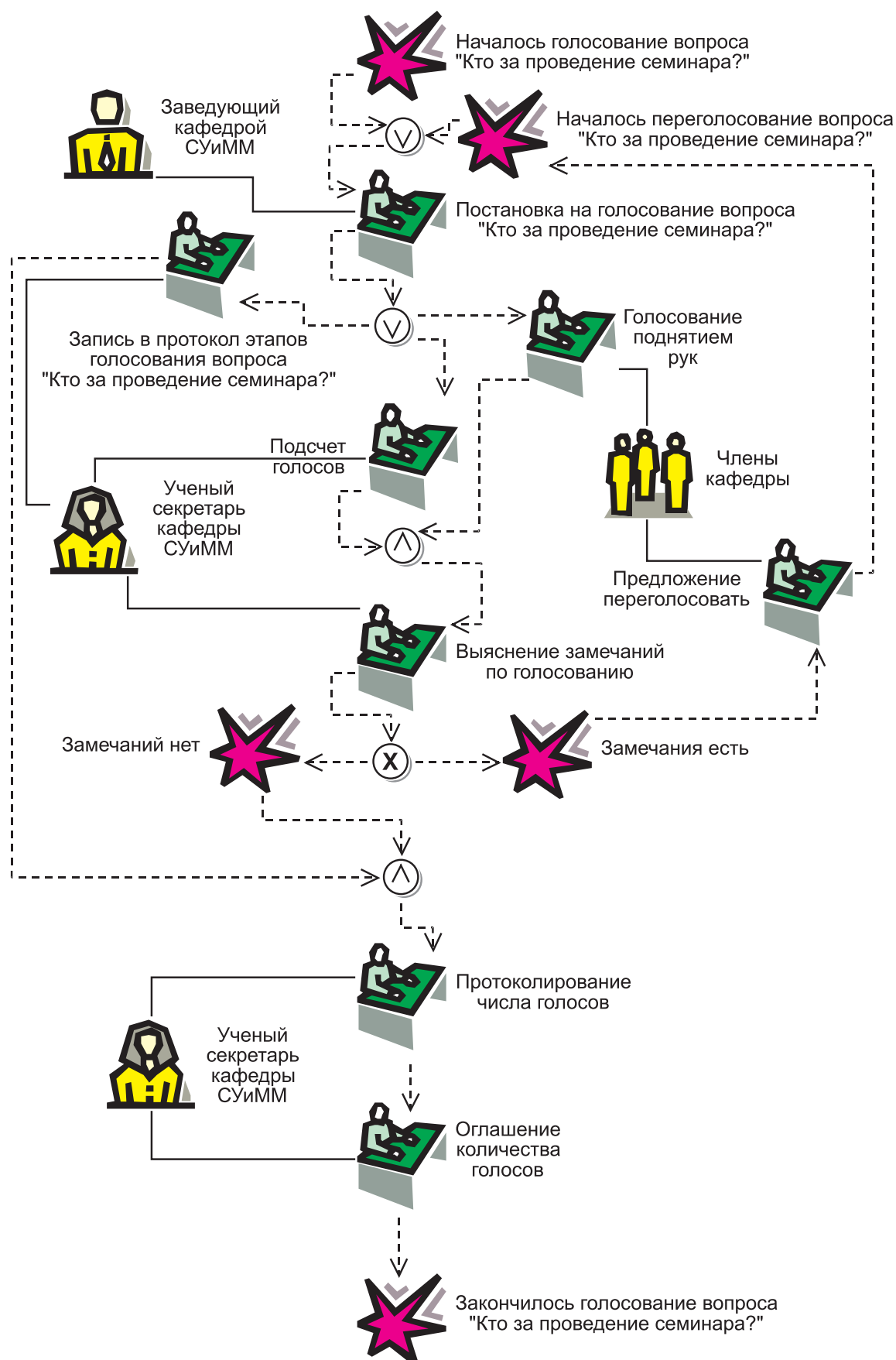


Рис.5.5.2 Экземпляр офисного процесса “Голосование вопроса “Кто за проведение семинара?” на заседании кафедры СУиММ”



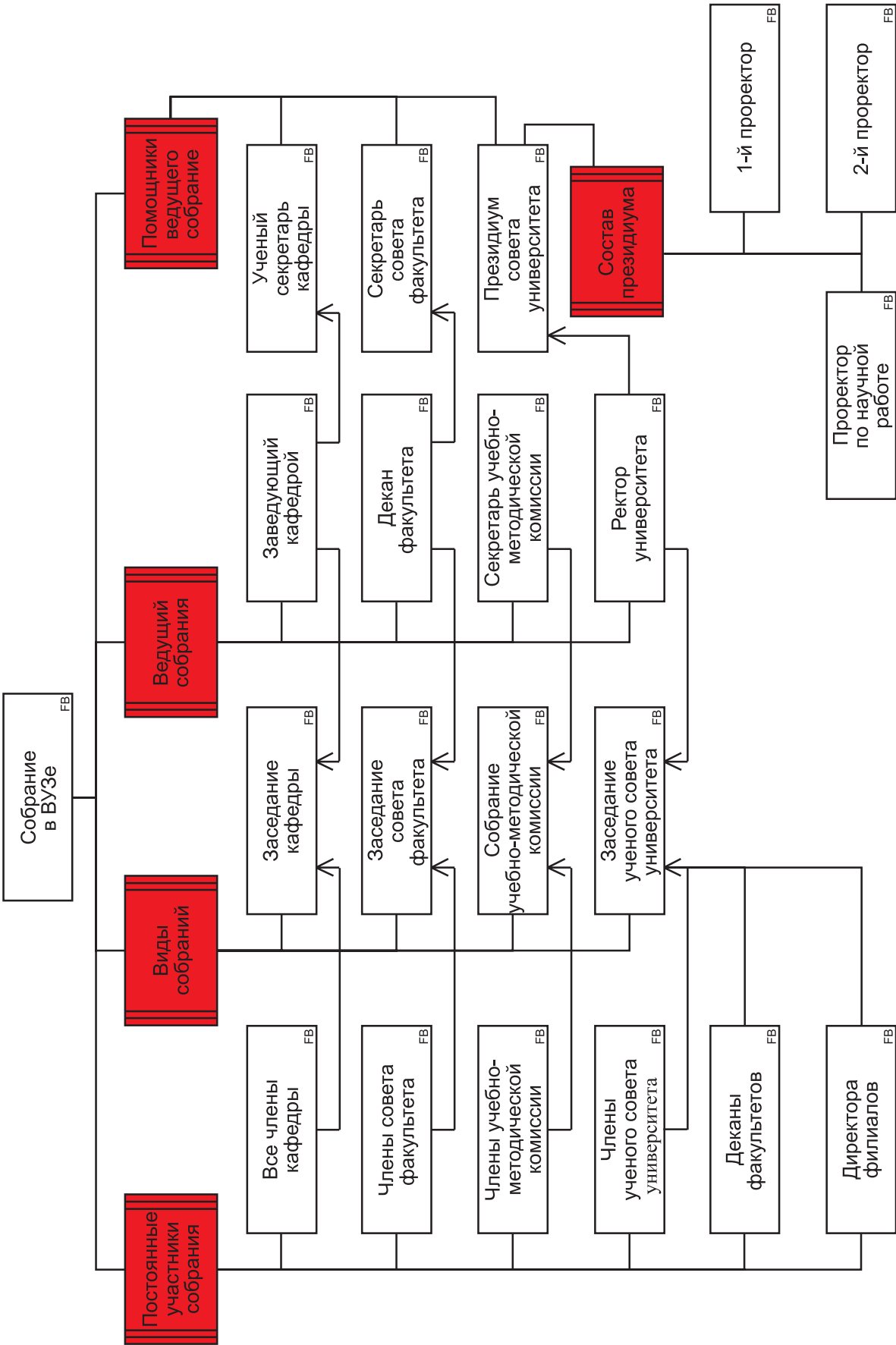


Рис.5.5.3 Диаграмма понятий, связанных с термином "Собрание в ВУЗе".

Уровни типа занимают важнейшее место в моделировании БП. Путем обработки частных на основе описания типового БП можно учитывать конкретные особенности отдельных экземпляров.

Представление экземпляров составляет 1-й уровень описания, а типов – 2-й уровень описания. На рис.5.5.4 приведено несколько примеров описаний 1-го и 2-го уровней. Здесь при описании понятия “заказ” осуществляется абстрагирование от специфических свойств (от номеров). В результате получены классы “оформленный заказ” и “готовый заказ”. На 2-м уровне осуществлено абстрагирование от свойств “оформленный” и “готовый” и создан из этого подмножества родительский класс “заказ”. Такая операция называется обобщением и обозначается символом “треугольник”.

При обобщении величины группируются в родительские классы. При этом экземпляры заказа 1-го уровня становятся и экземплярами класса “заказ”. Классу “заказ” приписывается свойство “состояние заказа”, что позволяет соотнести с каждым экземпляром класса состояние БП, описав заказ как “оформленный” или “готовый”. Материалы и изделия также обобщаются, становясь “детальными” и “ресурсами”. Таким образом, 2-й уровень содержит связанные с предметной областью классы описаний БП. Аналогично классы 2-го уровня путем абстрагирования от их частных объединяются в метаклассы на 3-м уровне описания, который называется метауровнем (рис.5.5.4). При этом классы 2-го уровня становятся экземплярами этих метаклассов. Например, класс “материальный выход” включает экземпляры “материал” и “деталь”, а также их обобщенное понятие “изделие”. Класс “информационные услуги” охватывает понятие “заказ” вместе с двумя его дочерними классами (“оформленный заказ” и “готовый заказ”), а также понятие “сертификат”. В роли элементов метаклассов могут выступать либо обобщенные классы 2-го уровня, либо их подклассы. На рис.5.5.4 показано формирование общего класса “тип объекта”, экземплярами которого являются все метаклассы.

На рис.5.5.5 изображен экземпляр модели типа “Размещение функции”, т. к. в ней конкретизированы основные объекты. На рис.5.5.6 эта же модель является типовой, т. к. в ней мы абстрагировались от конкретизации цели, ресурсов, оборудования и т. д. На рис.5.5.7 мы имеем метамодель “Размещения функции” и более того – это метамодель БП, т. к. из набора таких функций получается БП, но мы абстрагировались от последовательности и логики выполнения функций.

Если применить процедуру абстрагирования к моделям БП типа eEPC 2-го уровня описания, т. е. не об-

ращать внимание на такие частности, как количество, последовательность и логика следования функций, то получится общая АРИС-модель БП 3-го уровня, показанная на рис.5.5.7. На этом рисунке иллюстрируются общие классы, описывающие БП, и отношения между ними, обозначенные стрелками. Эти отношения тоже можно представить как классы (классы отношений). Модель на рис.5.5.7 не является исчерпывающей, но отражает важнейшие объекты, необходимые для описания БП. Метаклассы на 3-м уровне моделирования содержат объекты, необходимые для описания БП на 2-м уровне, т. е. эти объекты служат стандартными блоками или “кирпичиками” для описания приложений на 2-м уровне. С другой стороны, поскольку классы 2-го уровня используются в моделях 1-го уровня, то объекты 3-го уровня образуют инфраструктуру для описания конкретных БП.

Этот процесс абстрагирования можно продолжить. Группируя классы 3-го уровня и абстрагируясь от содержания модели, можно отнести полученные группировки к мета2-уровню. Следующий шаг в направлении абстрагирования приводит нас к рис.5.5.8, где все, что сосредоточено вокруг функции (главного объекта процессной модели) разнесено по 4-м видам моделей (выходам, данным, организационным структурам, функциональным деревьям и соответствующим целям).

Объекты 2-го уровня являются экземплярами метауровня, поэтому при построении типовых БП можно использовать только те объекты, классы которых определены на метауровне. Следовательно, все возможности проектирования (описания) БП заложены в метамодели.

Метамодель БП 3-го уровня описывает характерные классы и их взаимоотношения, с помощью которых можно моделировать процессы 2-го и 1-го уровней. Известно большое разнообразие классов и их семантических взаимоотношений. Структурирование этих классов приводит к структуре, называемой “Архитектура интегрированных информационных систем”.

Метамодель БП, приведенная на рис.5.5.7, объединяет следующие классы: контекстные данные, описывающие инфраструктуру БП, исходные и результирующие события, сообщения, функции, человеческий ресурс, технические ресурсы и компьютерные средства, прикладное программное обеспечение, материальный выход, выход в виде услуг и информационные услуги, финансовые ресурсы, организационные единицы, корпоративные цели. Многообразные отношения могут существовать и между самими классами. Для упрощения описания классы со сходными семантическими взаимоотношениями группируются в пять различ-

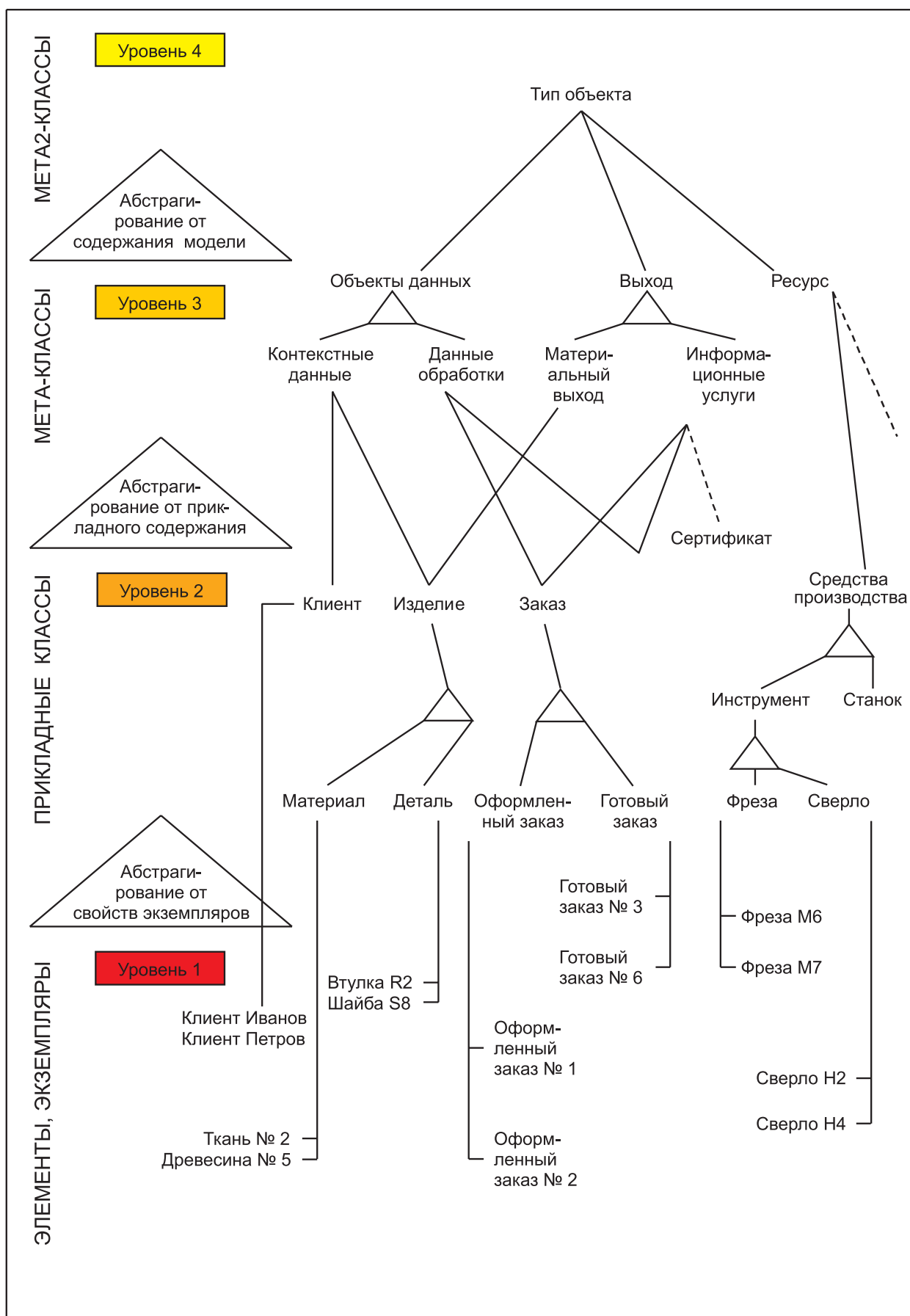


Рис.5.5.4 Уровни абстракции в моделировании

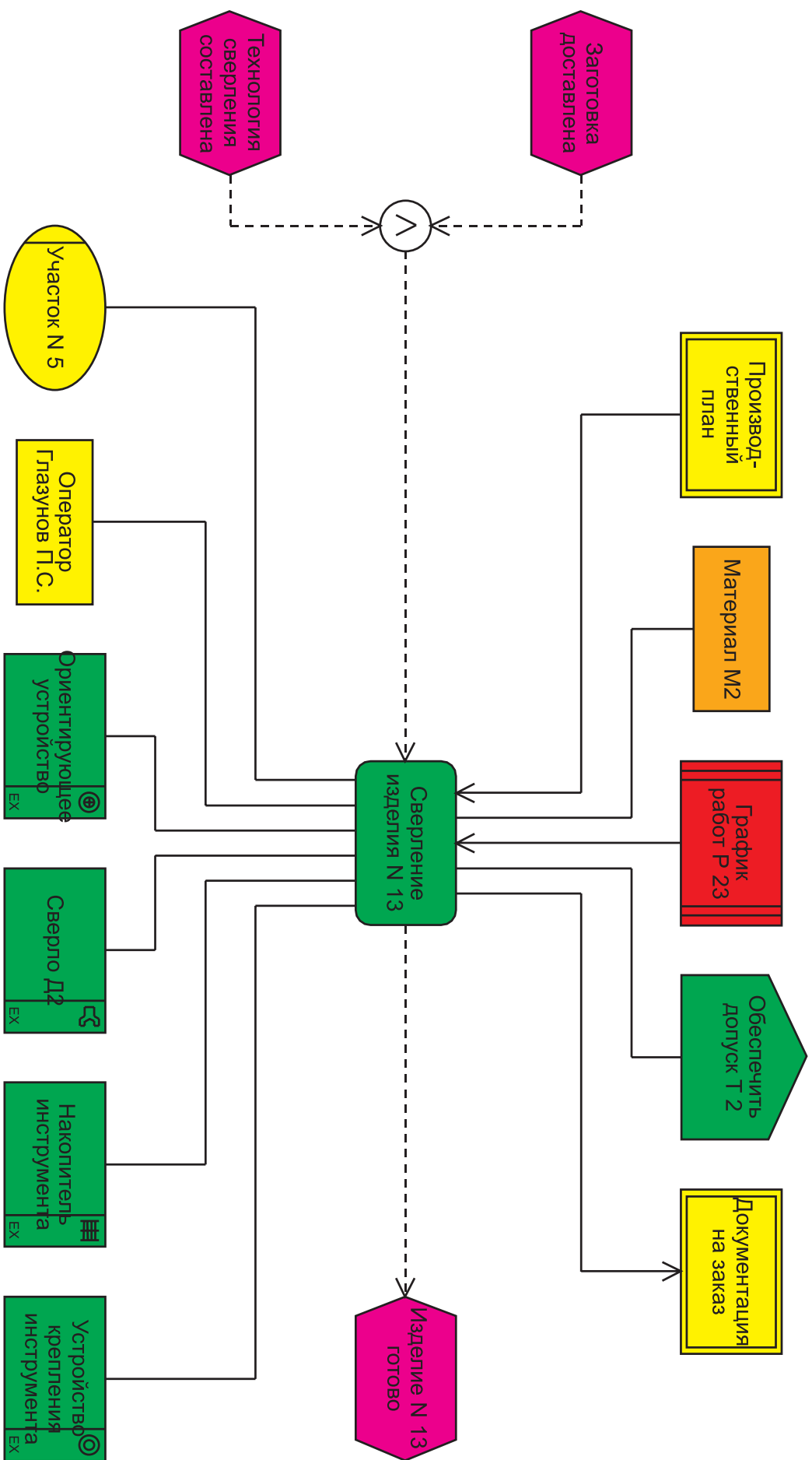


Рис.5.5.5 Экземпляр модели размещения конкретных объектов вокруг конкретной функции

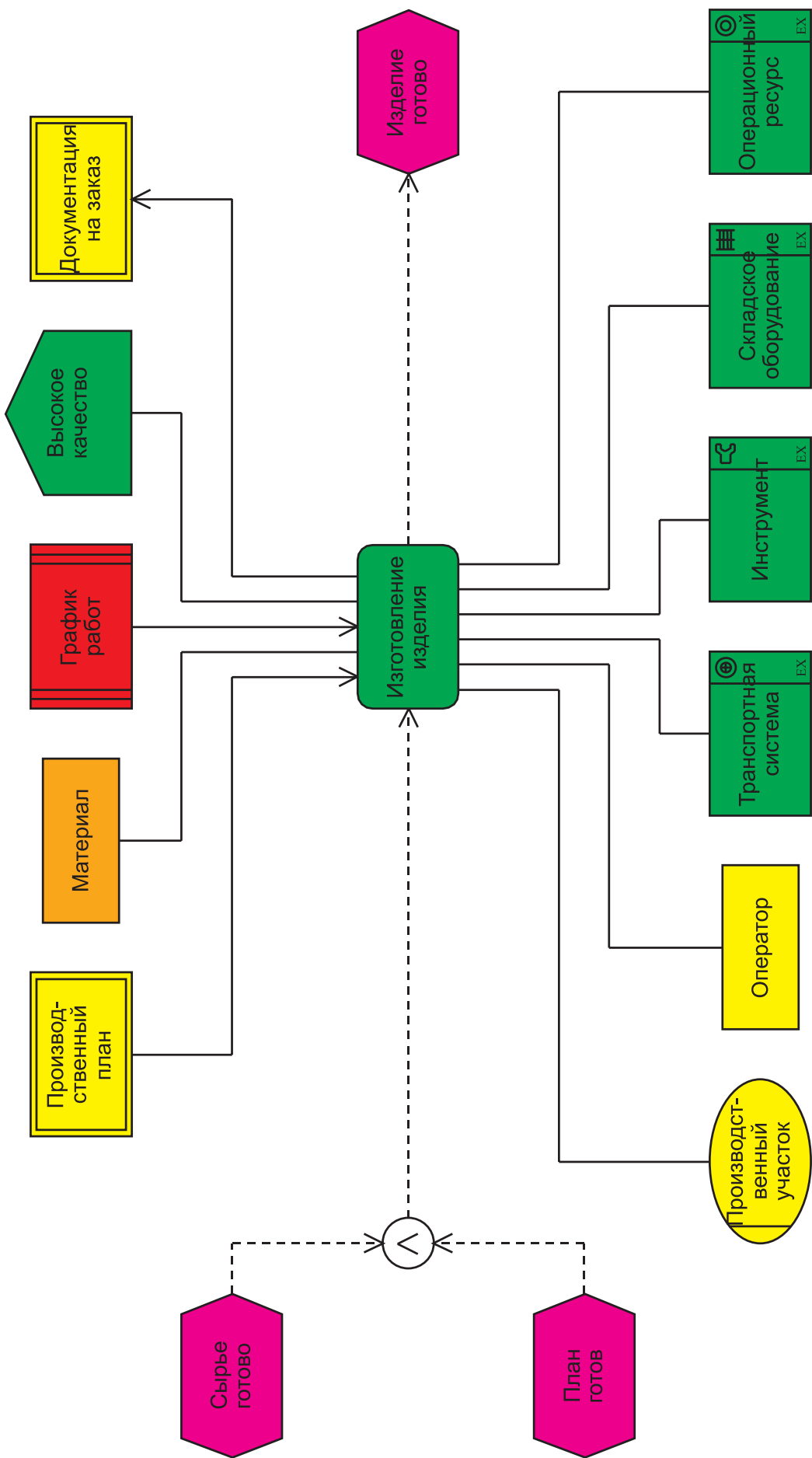


Рис.5.5.6 Типовая модель размещения объектов вокруг функции “Изготовление изделия”

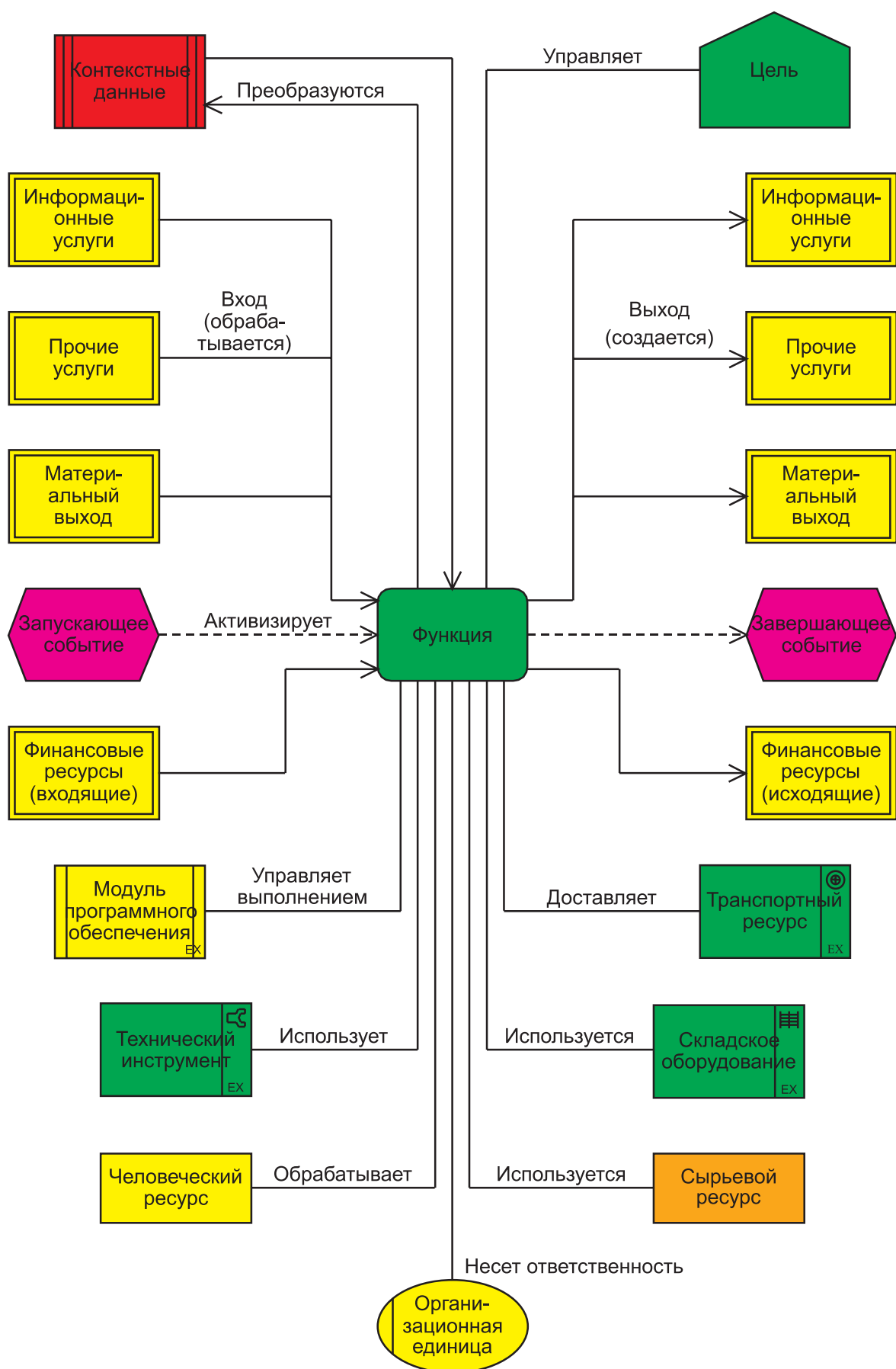


Рис.5.5.7 Общая ARIS-модель БП



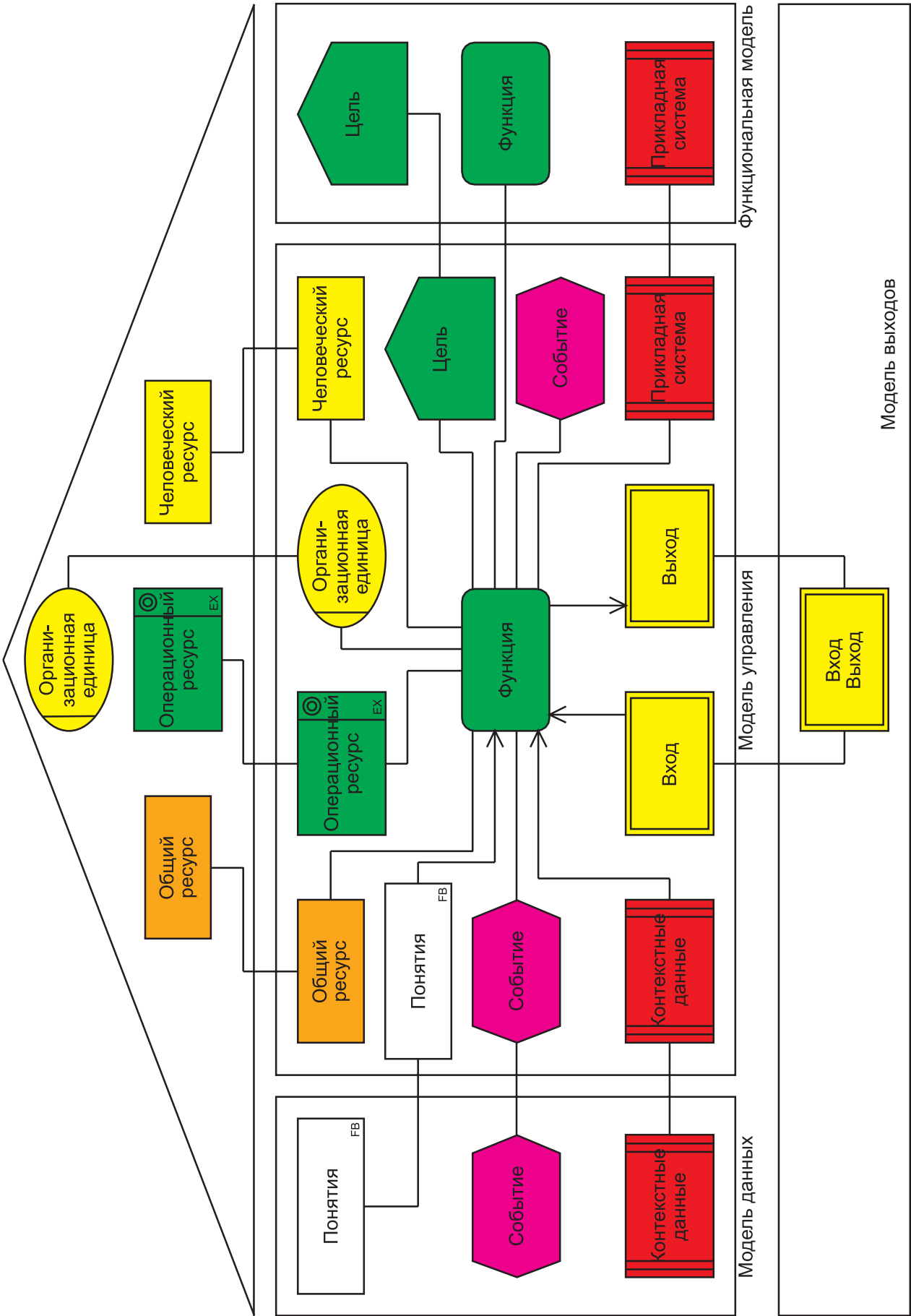


Рис.5.5.8 ARIS-модели

ных видов моделей (взглядов описаний). Это структурирование приводит к зданию АРИС на рис.5.5.8. Это позволяет в одном виде (взгляде) моделей рассмотреть связи классов (отнесенных к этому виду), избавляясь от необходимости сразу же учитывать весь комплекс взаимоотношений с моделями других видов (взглядов).

Если к этому добавить еще и трехэтапную трансформацию классов БП в объекты ИТ, рассматриваемых в фазах анализа требований, спецификации и реализации, то получим полную картину архитектуры интегрированных информационных систем.

Группировка классов и отношений между ними в виды моделей служит для структурирования БП, что позволяет устранить избыточность при многократном использовании объектов в модели БП. Модели различных видов строятся в соответствии с критерием “семантического корреляционного сходства”. В процессных моделях описываются отношения между отдельными моделями в рамках всего БП. Это позволяет отслеживать все двусторонние отношения между моделями различных видов, а также полностью описать БП. На рис.5.5.9 отражены уровни моделирования для всех пяти видов моделей.

Здание АРИС создает “каркас” для классификации описательных компонентов БП. Обсудим отдельные информационные “кирпичики” БП и отношения между ними. Рассмотрим метаязык, где фиксируются элементы общего БП, т.е. без привязки к конкретным БП. Основой для этого служит АРИС-модель БП, представленная на рис.5.5.8, в которой проанализируем отношения между информационными элементами. Для этого можно использовать унифицированный язык моделирования (UML). Полученное представление на рис.5.5.10 называется информационной моделью АРИС и описывает конструкцию БД, где можно хранить реальные модели. Организационные и функциональные модели, равно как и модели данных, выхода и управления, относящиеся к тому или иному приложению, рассматриваются как экземпляры БД, построенной в соответствии с информационной моделью. Такие БД называются репозиториями и являются ядром ИС.

Полная модель ИС АРИС подробно описывает классы и отношения между ними в модели мета-БП. Она описывает также все модели АРИС, охватывающие все фазы жизненного цикла. Такая модель может включать сотни классов и связей.

Информационная модель АРИС представляет собой схему репозитория для хранения соответствующих прикладных моделей. Данные, хранящиеся в репозитории, включают классы реальных приложений (напри-

мер, для сферы продаж или бухгалтерского учета).

На рис.5.5.11 сгруппированы четыре компонента мета-уровня с указанием их взаимосвязей: мета-БП АРИС, архитектура (здание АРИС), информационная модель АРИС и репозиторий АРИС. Концепция АРИС разработана на мета<sup>2</sup>-уровне, независимом от конкретной прикладной области. Информационная модель АРИС разрабатывается на мета-уровне. Иными словами, именно здесь определяются общие понятийные классы, описывающие БП, и отношения между ними. Реальные приложения моделируются на прикладном уровне, причем ИС для бизнеса обычно разрабатываются на уровне описания, что придает этим методам моделирования особую ценность.

Модели обычно создаются на уровне типов в соответствии с фазами проектирования БП. Это делает модели “неуязвимыми” к любым модификациям экземпляров, например, в тех случаях, когда в модель данных включаются новые изделия или из организационной модели удаляются отдельные сотрудники. Однако, если в определенном типе БП всегда присутствуют одни и те же экземпляры, то их можно использовать и для моделирования. Такие ситуации возникают, когда некие функции выполняются только определенным специалистом или многократно используется один и тот же экземпляр данных, или данная модель всегда связана с конкретной организационной единицей, а не с ее типом. Именно по этой причине в некоторых моделях уровни экземпляров и уровни типов объединяются.

Хранение и управление всеми моделями АРИС осуществляется на мета-уровне. Это делает их независимыми от конкретных методов, поскольку все описания, специфичные для того или иного метода и модели, являются экземплярами этого общего типа объекта. Таким образом, в АРИС можно включать новые методы моделирования, причем, как правило, без каких-либо изменений в программе.

Класс ТИП ОБЪЕКТА на мета<sup>2</sup>-уровне использует в качестве экземпляров объекты мета-уровня, например, типы функций, типы организационных единиц, типы выходов и т. д. К каждому объекту моделирования можно применять те же операции (создание, удаление, отбраживание, перемещение и представление в виде иерархии), которые определяются в классе ТИП ОБЪЕКТА. Если посредством новых методов вводятся новые объекты моделирования, то они рассматриваются как новые экземпляры данного класса, т.е. представляются соответствующими строками в таблице типов объектов.

В том случае, когда создаются модели экземпляров (например, для приложений workflow), интерпретация



Рис.5.5.9. Уровни АРИС-моделирования

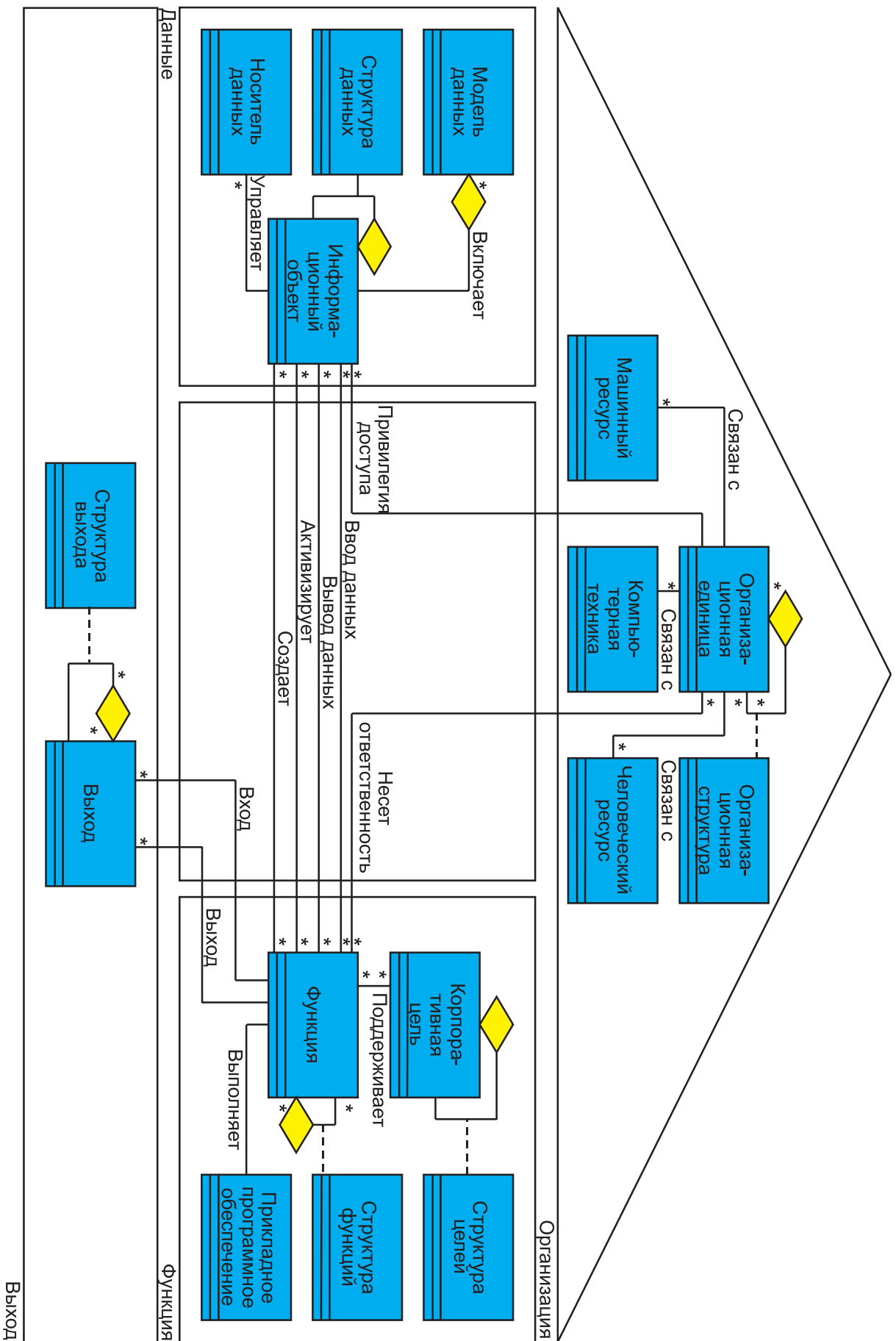


Рис.5.5.10 Информационная модель АРИС

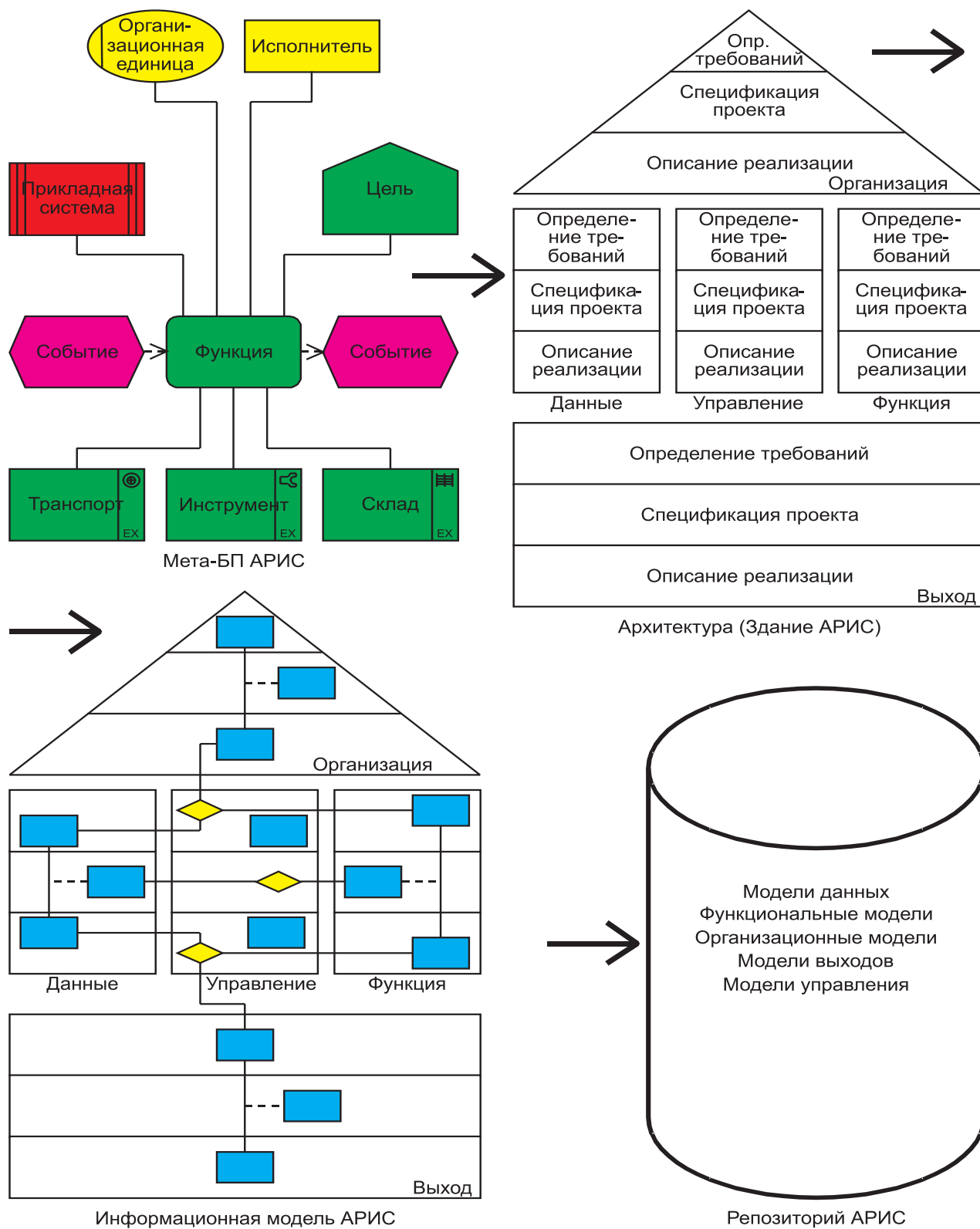


Рис.5.5.11 Взаимосвязь компонент метауровня



мета- и мета2-моделей АРИС расширяется. В структуре метамодели меняются допустимые экземпляры. Мета2-модель класса ТИП ОБЪЕКТА включает различные описания экземпляров прикладных объектов, а также реальные экземпляры этих прикладных объектов. Таким образом, в таблицы вводятся не только типы приложений, но и их экземпляры. То же справедливо для связи СОЕДИНЕНИЕ и для других связей.

Каждая модель в АРИС логически хранится в нескольких больших таблицах. Эти таблицы реализуются в объектно-ориентированной БД (РОЕТ), обеспечивая более детальные и, следовательно, более эффективные, с точки зрения производительности, структуры доступа (например, к комплексным моделям).

### 5.6. Детализация и структуризация моделей

АРИС-моделирование систем и процессов можно осуществлять *сверху вниз* (нисходящее моделирование путем декомпозиции и анализа), или *снизу вверх* (восходящее моделирование путем агрегации и синтеза), или одновременно снизу-вверх и сверху-вниз.

Следует заметить, что познание объективного мира осуществляется в едином процессе анализа и синтеза. Анализ – мысленное или реальное разделение целого на части (“уже разрезание яблока есть начало анализа”). Анализ всегда связан с синтезом, т. е. процессом, в котором мысленно или реально разрозненное, расчлененное соединяется в некоторое целое (целостность). Синтез является не только противоположностью анализа, но и необходимым условием самого анализа (без анализа нет синтеза и наоборот). Он понимается как мысленное или реальное соединение частей в единое целое. Причем, это соединение является не суммированием, а *интеграцией*, так как при синтезе создается нечто новое, с новыми свойствами, отсутствующими у составных частей.

При моделировании используют как аналитический, так и синтетический подход. Аналитический подход требует расчленения, углубления понимания сущности, неизбежного разрыва связей в предметах, явлениях, системах. Синтетический подход, напротив, требует выявления многообразия связей, формирования целостности и системности.

При моделировании снизу вверх начинают с построения простейших моделей, которые состоят из элементарных объектов. На рис.5.6.1-5.6.4 представлены четыре вида простейших моделей, т. к. все их объекты элементарны, т. е. далее не делимы.

Моделирование сверху вниз начинается с построения комплексных моделей, в которых большинство объектов являются сложными моделями и нуждаются в дальнейшей детализации. Степени детализации подразделяются на несколько (как правило, до десяти) ярусов (уровней). На верхнем уровне представлены комплексные модели (супер-модели), на нижнем – элементарные (далее не делимые) функции, организационные единицы, понятия, элементы данных и т. п. На рис.5.6.5-5.6.8 представлены комплексные модели, в которых у большинства объектов справа внизу указана пиктограмма дальнейшей детализации.

Диаграммы верхних уровней определяют те БП системы, в которых отражена связь моделируемой системы с надсистемой, где показаны все внешние входы и выходы. Далее эти БП детализируются вплоть до диаграмм нижнего уровня. Такая функциональная декомпозиция приводит к созданию многоуровневой иерархии диаграмм. На нижнем уровне декомпозиции все объекты (функции, организационные единицы, данные) становятся элементарными, т. е. не подлежащими дальнейшей детализации. Число уровней представления моделей – 5-7, иначе могут возникнуть проблемы при анализе. Когда дальнейшая детализация, например, логических функций перестает быть полезной, то переходят к выражению внутренней логики процессов при помощи миниспецификаций, т. е. алгоритмов преобразования входных потоков в выходные.

Очень важное значение при построении комплексной модели предприятия имеет *модель верхнего уровня*, которая представляет собой модель с наибольшим уровнем абстрагирования. Она необходима для понимания взаимодействия между БП, а также для организации процессного управления. В названии процесса верхнего уровня отражается основная деятельность предприятия. Второй уровень декомпозиции модели верхнего уровня может состоять из следующих БП: выполнить текущую деятельность, анализировать результаты выполнения текущей деятельности, формировать управляющее воздействие по изменению текущей деятельности. Это позволяет отделить текущую деятельность владельцев БП от их управленческой деятельности на уровне каждого БП.

Моделирование сложных систем и БП лучше всего проводить по обеим технологиям, т. е. путем декомпозиции (анализа) и агрегации (синтеза). После нескольких итераций примерно в середине иерархии построенных моделей они (модели) начнут совпадать друг с другом (согласно принципу сравнимости моделей). Это верный признак и хороший показатель качественного

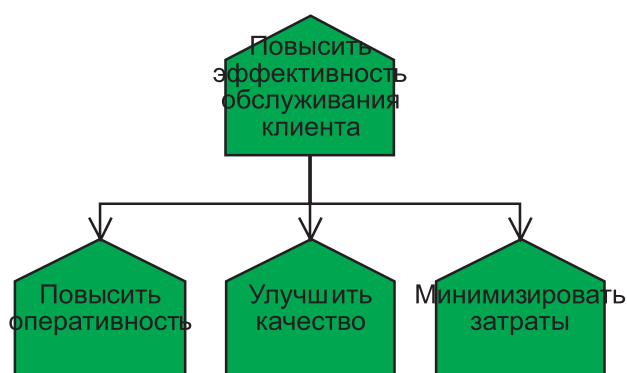


Рис.5.6.1 Фрагмент дерева целей

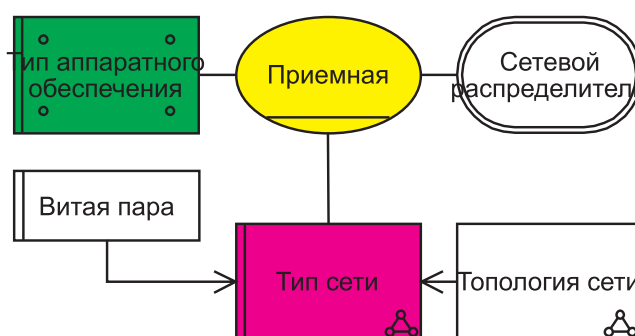


Рис. 5.6.2 Диаграмма топологии сети

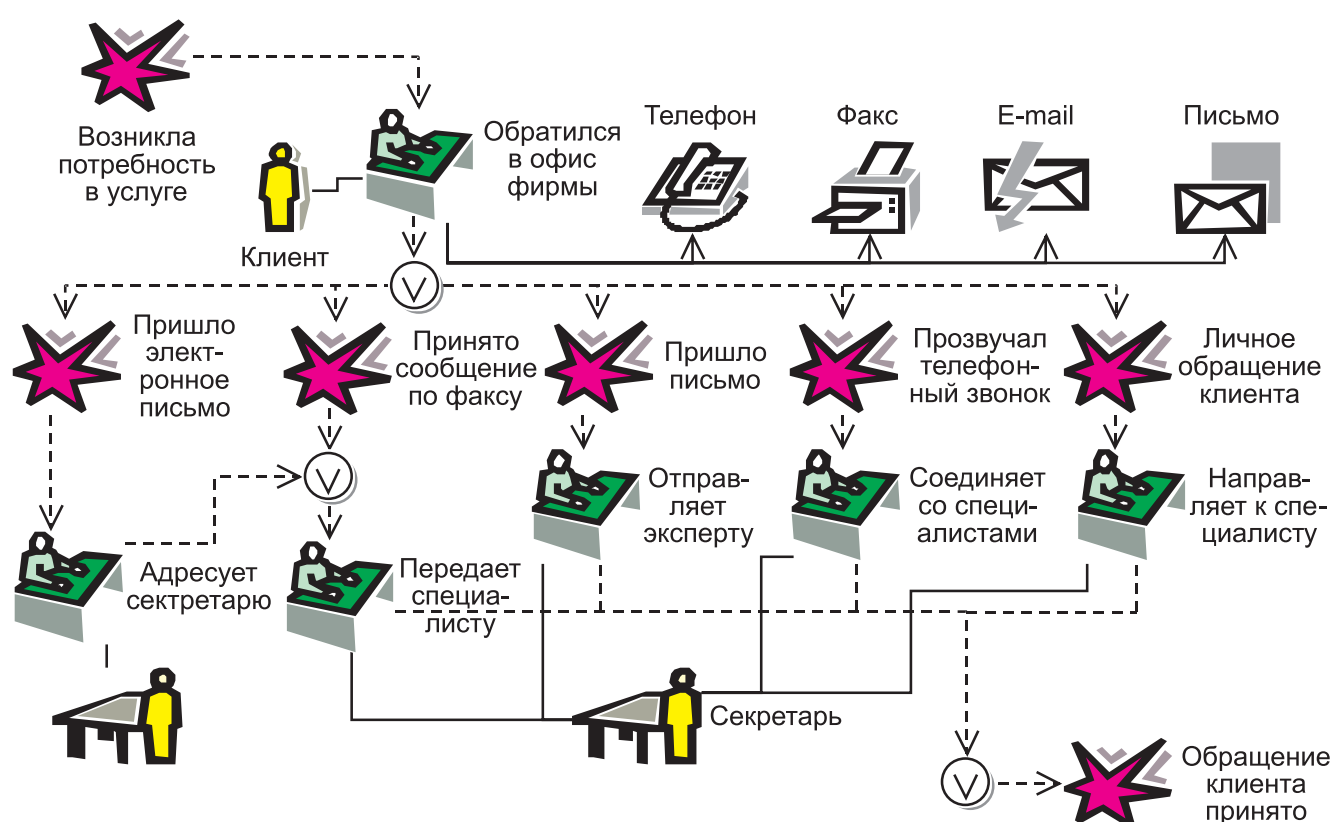


Рис. 5.6.3 Фрагмент процесса приема обращения клиента

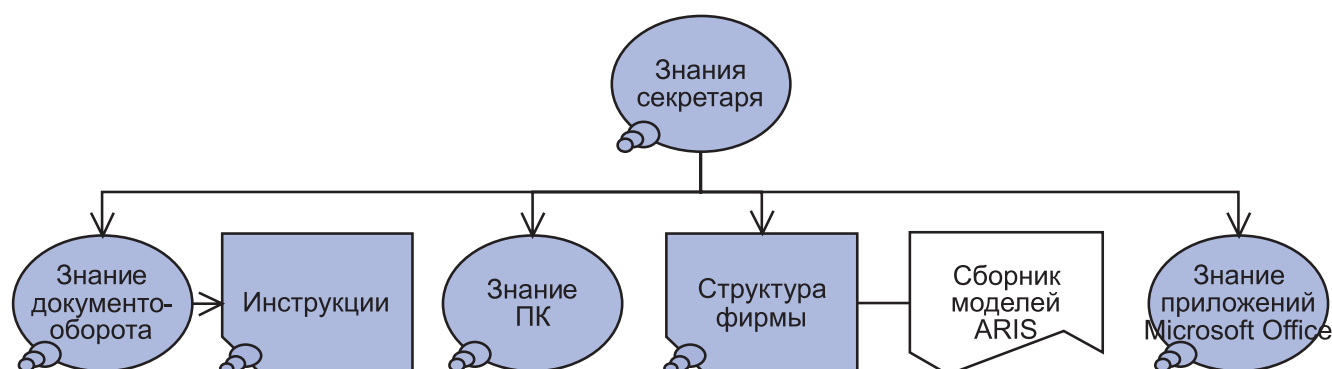


Рис.5.6.4 Диаграмма структуры знаний секретаря фирмы

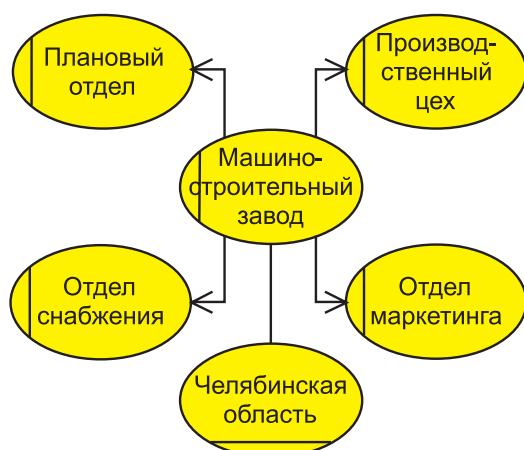


Рис. 5.6.5. Фрагмент организационной диаграммы предприятия

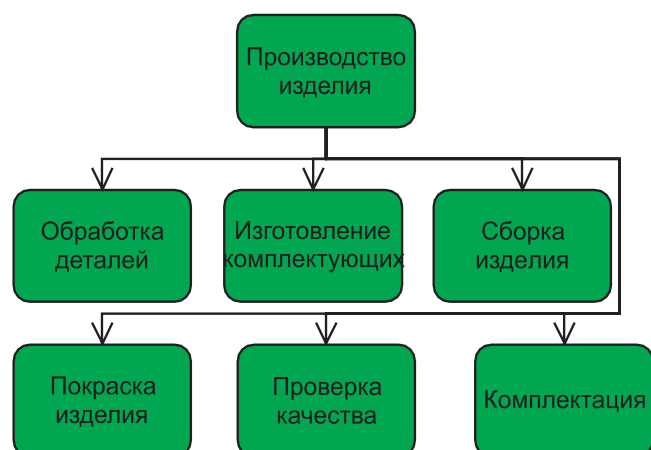


Рис. 5.6.6. Процессно-ориентированное дерево функции "Производство изделия"

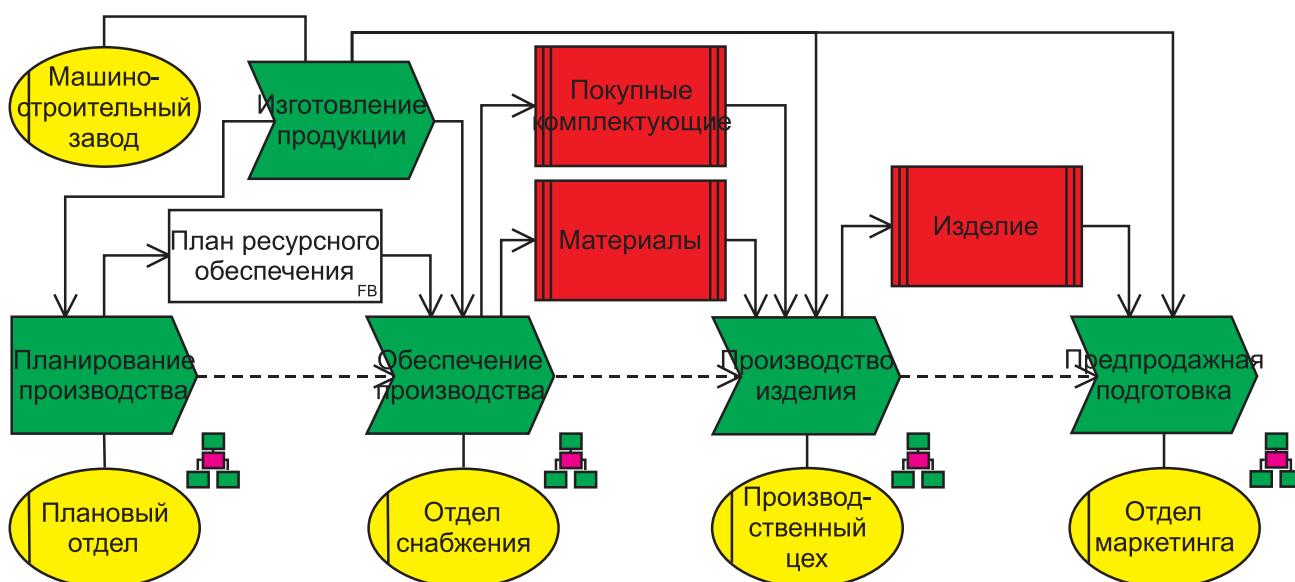


Рис. 5.6.7. Диаграмма добавленной стоимости для типового процесса производства



Рис.5.6.8 Фрагмент модели составляющих понятие "План ресурсного обеспечения"

моделирования. Чем больше совпадающих моделей, тем больше уверенность в правильности выбранного подхода и эффективности проделанной работы.

Детализация и структуризация позволяют описывать различные сложные системы. Любую модель можно построить в соответствии с выбранными критериями бизнеса и наглядно представить в виде мозаики, где отдельные элементы складываются в цельную картину.

Методика и способы хорошего описания системы, в том числе применение детализации и структуризации, пока отсутствуют в литературе. Но кое-чему можно научиться на хороших образцах, где в первую очередь обращают внимание на организацию или структуризацию информации. В основе хорошего описания всегда лежит иерархическая декомпозиция. Общий совет – разбивать объект на части, но частей должно быть «не слишком много и не слишком мало» (5-7). С каждой частью поступать аналогично, пока дальнейшее разбиение станет уже «не нужным для ясного понимания». Сначала даем предварительное описание, в котором предмет описывается в общих чертах, с упором на интуицию и содержательную сторону, а потом уточняющее описание, в котором предварительное описание дополняется, уточняется и детализируется.

Известны средства «укрупнения» описания вообще, включая обрабатываемые данные, связь между данными и применяемыми к ним действиями и т. д. Описывая процесс или предметную область, особенно большую и сложную, естественно попытаться представить ее в виде относительно крупных и осмысленных «блоков», подобно тому, как картину естественно сначала набросать крупными штрихами или мазками. Каждому блоку соответствует емкая смысловая единица, понятие или супер-процесс. Введение такой «крупноблочности» придает описанию более или менее упорядоченную и прозрачную структуру.

Этот прием описания сложной системы представляет собой вариант абстракции, который позволяет легче охватить область, придать ей обозримую структуру, уменьшить сложность ее восприятия и осмысления. К выделенному понятию или подпроцессу иногда можно, в свою очередь, применить тот же прием, и тогда мы получим несколько уровней описания с разной степенью детализации или использующих разные средства. Если описания на разных уровнях или построенные разными средствами претендуют на то, что они по-разному представляют «одно и то же», то это должно быть достаточно точно определено и аргументировано.

Количество понятий в сложной системе очень велико. Сгруппировав их в классы (кластеры), обладаю-

щие внутренним единством, можно в какой-то степени их организовать. Организовать что-либо или кого-либо – значит ввести некоторый порядок или систему, полезную для достижения какой-то цели.

При описании рекомендуется разделять аспекты и структурировать объекты и процессы по горизонтали и вертикали. При упорядочении, систематизации и осмыслении круга понятий, относящихся к процессу или системе, очень полезный прием состоит в выделении частей-аспектов или групп (кластеров) объектов. Принципы выделения или разделения предмета на части, аспекты, группы в конкретных ситуациях могут быть самыми разными. Общее в них только то, что выделение или разделение должно быть осмысленным. В результате всего этого аморфная вначале предметная область разделяется на осмысленные блоки и, благодаря этому, становится более обозримой и понятной.

Выделение аспектов, частей, групп, имеющих самостоятельное значение, дает возможность гибко комбинировать их в соответствии с разными целями. Одна из центральных идей АРИС в том и состоит, что надо тщательно и осмысленно разделить понятийные средства описания, чтобы потом их можно было соединять и комбинировать любыми нужными способами. Хорошее соединение (синтез) требует хорошего разделения (анализа)!

Выделение (разделение) обычно сопровождается установлением или выявлением связей между выделяемыми частями, аспектами, группами и внутри них. В итоге получается некоторая структура рассматриваемой системы, области или объекта, т. е. речь идет о процессе структуризации. Выделяют два типа структуризации описания – вертикальную и горизонтальную. Первая связана с разделением описания (и процесса описания) на уровни и дает, так сказать, уровневую структуру описания. Вторая дает структуру описания на одном выделенном уровне.

## 5.7. Управление знаниями

Корпоративные знания содержат информацию (в том числе, ноу-хау) о продуктах, технологиях, рабочих процедурах и правилах, а также индивидуальные знания и умения каждого конкретного работника. Одна из первоочередных задач управления знаниями заключается в документировании, хранении, использовании и расширении этой информации. Важный шаг к организации управления знаниями – создание их хранилища. В АРИС имеются средства для представления хранилищ структурированной корпоративной информации о технологических процедурах и ресурсах, а также о ноу-хау,

которым обладают конкретные работники. На уровне модели данных знания фиксируются в виде текстовых, голосовых, графических и видео-документов.

Различных создателей знаний по отношению к компании (индивидуумы, группы, сама компания, группа компаний, альянсы компаний) можно описать на уровне организационной модели, а затем связать их с типами знаний в других моделях. Таким образом, АРИС становится инфраструктурой для “корпоративной памяти” (“хранилища знаний”). Хранилища знаний выявляют и устраняют дефицит знаний, неиспользуемые знания, недостаточную прозрачность знаний и неэффективное распространение знаний.

Из-за отсутствия определенной информации часто допускаются дорогостоящие ошибки. Например, критически важные знания оказываются утраченными в связи с уходом сотрудников. Децентрализация компании влечет за собой и децентрализацию знаний, поэтому одна из важнейших проблем – обеспечить доступность ключевой информации в масштабах всего предприятия. В противном случае в одних и тех же ситуациях приходится каждый раз заново “изобретать велосипед”.

С целью обеспечения управления одним из важнейших экономических ресурсов (информацией) в компаниях часто учреждается новая должность – директор по знаниям, в функции которого входят разработка, мониторинг и совершенствование стратегий, процессов, организационных структур и технологий для обработки корпоративной информации. Сбор, отображение, представление и использование знаний осуществляются также в рамках БП, что является важнейшей предпосылкой для целенаправленного управления знаниями.

Для совершенствования БП необходимо эффективное использование накопленных и оформленных в виде документации знаний. Процедурная модель для совершенствования использования корпоративных знаний включает следующие фазы: стратегическое планирование знаний, моделирование обработки знаний “как есть”, анализ состояния “как есть”, целевые модели обработки знаний, организационно-кадровая реализация и реализация средствами ИТ, комплексная реализация. Рассмотрим подробно перечисленные фазы (этапы) реинжиниринга процессов использования знаний.

Одна из важнейших задач на этапе *стратегического планирования знаний* – выяснить, какие знания имеют ключевое значение для предприятия. Стратегическое планирование знаний опирается на стратегические корпоративные цели. Основное назначение данной фазы заключается в том, чтобы определить, каким образом управление знаниями может способствовать достиже-

нию этих целей. Например, компания может ставить перед собой стратегическую цель: занять или сохранить лидирующее положение в качестве поставщика технологий. Проект управления знаниями должен обеспечивать сбор и учет внешней информации об этих технологиях. Необходима также координация и, возможно, активизация внутренних научно-исследовательских и опытно-конструкторских разработок в этой области.

На этапе *моделирования обработки знаний “как есть”* анализируются существующие способы обработки знаний. Определяется, где размещаются корпоративные знания, как и где они создаются и каким образом используются.

Для того, чтобы описать процесс обработки знаний, важно построить модель, отражающую, какие категории знаний обрабатываются, кто какими знаниями обладает, какие знания требуются для конкретных функций, где эти знания создаются или откуда их можно получить. Для этого используются структурные диаграммы знаний, карты знаний и знания в функциональных потоках.

*Структурные диаграммы знаний* описывают актуальные знания для предприятия. Следует различать подразумеваемые знания, которыми обладают сотрудники, и документированные знания.

*Карты знаний* – это модели, представляющие сотрудников или организационные единицы, которые обладают конкретными знаниями. Они могут отражать также степень полноты знаний. Карты знаний позволяют выявить области неполных знаний и монопольных носителей знаний.

*Знания в функциональных потоках* содержат дополнительную информацию об использовании знаний в моделях существующих БП, где показывается, например, какие знания необходимы для реализации определенной функции или какие знания документально фиксируются при выполнении функции.

В атрибутах объектов знаний отражаются способы представления и подготовки знаний и наличие или отсутствие мотивации для обмена знаниями.

В результате *анализа состояния знаний “как есть”* в существующих БП можно выявить недостатки (например, монополизацию или слабое использование существующих знаний) и потенциальные возможности повышения эффективности использования знаний.

Особые преимущества моделирования “как есть” заключаются в том, что оно позволяет документально зафиксировать, какими знаниями располагает предприятие, и где эти знания находятся. Соответствующая документация помогает быстро отыскать нужного человека для решения конкретных вопросов.

К слабым местам и областям возможного совер-



шенствования обработки знаний относятся стратегически важные знания, не охваченные предприятием; монопольное владение знаниями, приводящее к их утрате, например, в связи с уходом из компании сотрудников; неудовлетворенные потребности в знаниях; неиспользуемые знания; приобретение и описание одних и тех же знаний каждый раз заново; бесполезные или устаревшие знания сотрудников; проблемы приобретения и использования знаний, затрудняющие их распространение; отсутствие согласованности информационных и коммуникационных технологий, предназначенных для обработки знаний.

В целевых моделях обработки знаний оптимизируются существующие БП и разрабатываются новые (целевые) процессы сбора, подготовки и распространения знаний. В БП вносятся изменения, направленные на улучшение обработки знаний. Например, вводятся дополнительные функции, которые документируют приобретенный опыт и передают его другим сотрудникам. Здесь можно описать функции, предназначенные для обновления, структурирования и распространения знаний, а также для удаления устаревших знаний. Можно, например, для отдела продаж описать общекорпоративный процесс сбора данных о требованиях и пожеланиях клиентов.

На этапе *организационно-кадровой реализации* намеченных изменений необходимы соответствующие организационные и кадровые мероприятия (например, обучение персонала), а также использование возможностей информационных и коммуникационных техно-

логий. Профильные знания используются в курсах повышения квалификации персонала или при приеме на работу новых сотрудников.

Этап *реализации средствами ИТ* используется для интегрирования разных систем и обеспечения доступа к содержанию знаний с точки зрения их согласованного распространения (доставки). Здесь необходима единая структура в виде информационных и коммуникационных систем. Сюда включаются и такие аспекты, как структурирование содержания, описание пользовательских интерфейсов, внедрение специальных служб типа дискуссионных групп или подписки на информационные услуги.

Для *комплексной реализации* следует обучить персонал, протестировать новые процедуры и постоянно совершенствовать их. Здесь предполагается проведение курсов обучения, обеспечение соответствия персонала квалификационным требованиям, подготовка и контроль модификации БП и организационных структур, а также внедрение информационных и коммуникационных технологий.

В заключение отметим, что основу корпоративных знаний составляют описания БП. Систематический сбор, хранение и ведение (сопровождение) ноу-хау БП в репозитории воплощаются в так называемом хранилище процессов. К сожалению, в российских компаниях знания не документированы, кроме, быть может, еженедельных меню столовых предприятий.



## 6. ОПИСАНИЕ ПРОГРАММНОЙ СИСТЕМЫ ARIS

*ARIS-методология послужила основанием для разработки семейства инструментальных программных средств ARIS. Разработчик ARIS – компания IDS Scheer AG (Германия). В этой главе рассматриваются основы ARIS.*

### 6.1. Структура и возможности ARIS

Вербальное описание – это нудная проза и многословие там, где достаточно одной диаграммы. Ясность изложения можно значительно повысить, используя графические изображения (диаграммы). ARIS 5.0 позволяет строить 130 различных типов диаграмм, чем обеспечивает всестороннее описание исследуемой системы и ее процессов. Но для того, чтобы описать систему на 80%, достаточно построить не более 20% основных (базовых) типов моделей (диаграмм), например, диаграмму цепочки добавленной стоимости (Value Added Chain Diagram), организационную структуру (Organizational Chart), дерево функций (Function Tree), событийно-управляемый функциональный поток (eEPC, extended Event-Driven Process Chain), диаграмму размещения функции (FAD, Function Allocation Diagram), модель сущность-отношение (ERM, Entity Relationship Model), офисный (Office Process) или производственный процессы (Industrial Process). ARIS также позволяет документировать сетевые архитектуры, компоненты технических средств, логические структуры модулей, БД и многое другое. На рис.6.1.1 для каждого вида моделирования представлены некоторые из основных типов моделей.

Начиная с 5-й версии, ARIS является инструментальным средством не только моделирования и анализа, но и оптимизации БП. Это, в частности, достигается за счет появления в нем нового набора инструментов – Process Performance Manager, способного собирать информацию о БП, оценивать его состояние (графически отражая результаты в различном виде) и сравнивать с другими БП. Среда ARIS предназначена для системных аналитиков предприятия, которые занимаются созданием, анализом и оценкой своей деятельности в терминах процессного подхода.

ARIS позволяет следующее: описывать, анализировать и оптимизировать БП, сравнивать альтернативные варианты БП, строить модели организационной структуры, модели данных, функциональные модели, модели процессов (управления), создавать должностные инструкции и технологические карты БП методом документирования моделей, осуществлять имитационное (динамическое) моделирование БП, оценивать стоимость БП (на основе пооперационного стоимостно-

го анализа), искать и анализировать затратные центры, описывать БП на разном уровне детализации, управлять цепочками поставок (Supply Chain Management), проектировать сценарии электронного бизнеса (e-Business), управлять знаниями предприятия, использовать модели-прототипы для описания предприятия, формулировать и выдавать отчеты в требуемом виде, значительно ускорять внедрение системы SAP R/3 при использовании интерфейса между ARIS и модулем AcceleratedSAP, организовывать публикацию моделей и их совместное использование в Internet/Intranet, поддерживать технологию OLE, организовывать работу в многопользовательской среде (клиент-сервер), интегрироваться с программными системами Staffware (WorkFlow), Lotus Notes (GroupWare), Microsoft Project, ER/win, Designer/2000 и др.

Два основных аспекта ARIS – это организация информации и работа с информацией. К первому относятся типы и структура информационных объектов, виды связей, группировка в классы и семейства. Ко второму – команды, обеспечивающие диалог с системой, поиск объектов, изменение состояния БД, получение информации об организации самой системы.

Понятия модель, объект, отношение и связь являются общими для всех видов моделирования. Модель состоит из объектов. Между объектами модели существуют определенные типы отношений. Объекты в модели могут быть связаны с детализирующими моделями. Модели, объекты, отношения и связи, равно как и все остальные сущности (серверы, БД, группы и др.), имеют свойства и атрибуты, которые размещаются в таблицах и позволяют детально описать БП.

На рис.6.1.2 изображен фрагмент структурной диаграммы программного комплекса ARIS, где справа или под организационной единицей приведен текст с кратким описанием ее назначения. Из этой диаграммы видно, что основные функциональные модули ARIS-комплекса состоят из четырех базовых компонентов: проводника (*Explorer*), дизайнера (*Designer*), таблиц (*Tables*) и мастеров (*Wizards*).

Проводник (*Explorer*) включает в себя основные управляющие функции для администрирования моделей, объектов и другой информации и аналогичен проводнику в Windows. ARIS Explorer, в основном, используется для администрирования сущностей в ARIS-ре-

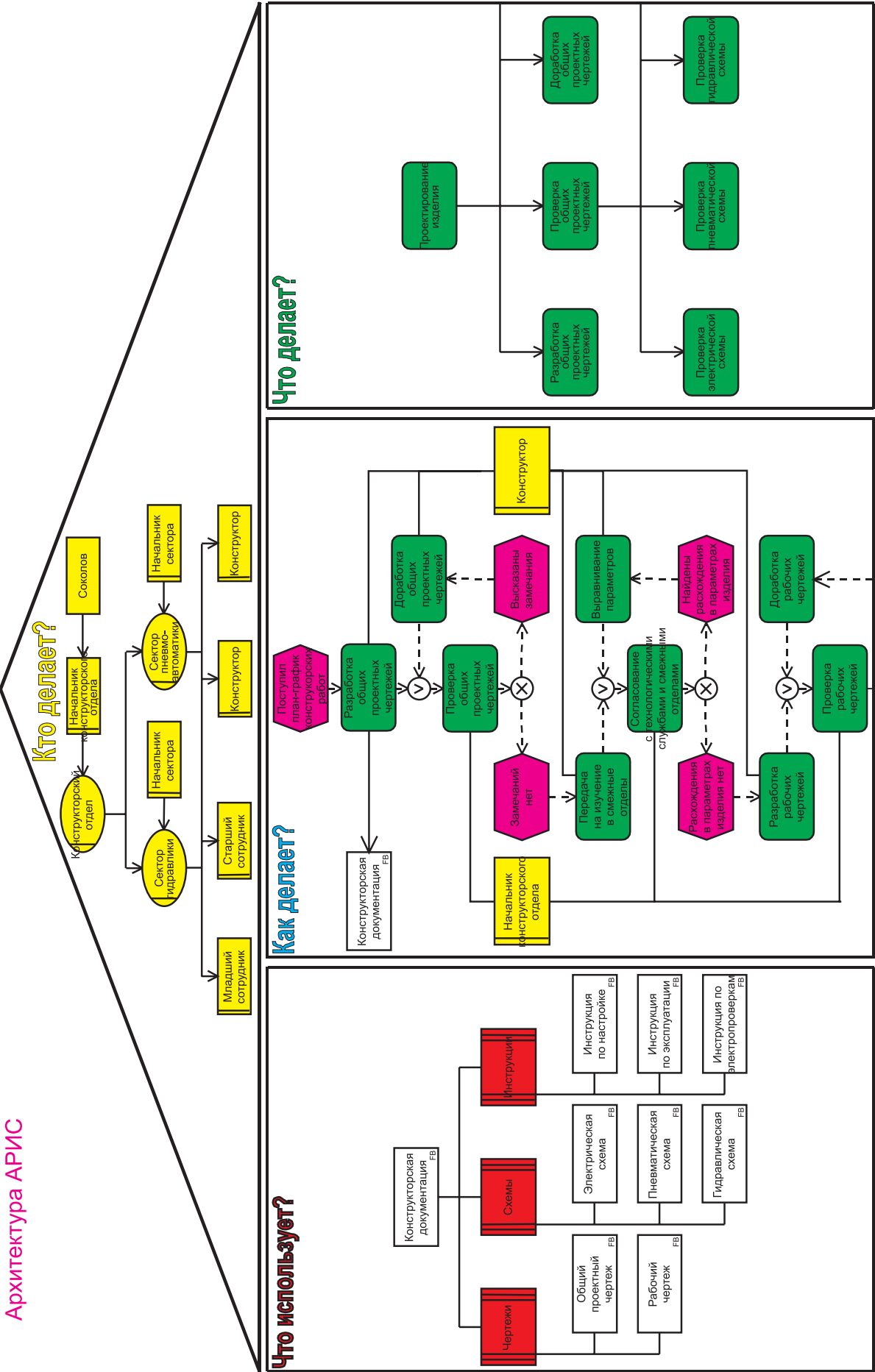


Рис.6.1.1 Виды и типы моделей в ARIS 5.0 с Easy Filter

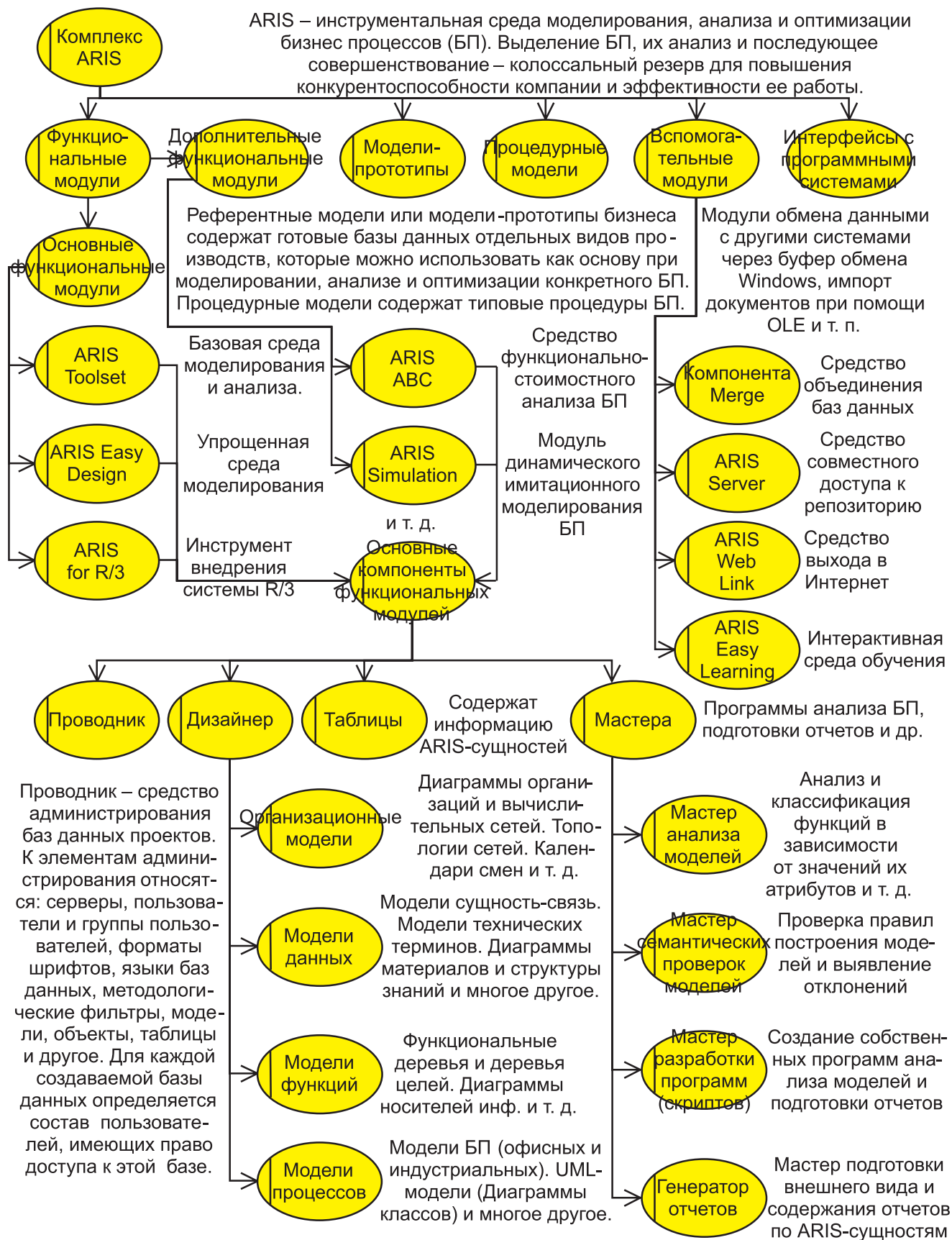


Рис.6.1.2 Состав программного комплекса в ARIS

позитории, т. е. серверов, БД, пользователей и групп пользователей, форматов шрифтов, языков, фильтров методов, моделей, групп моделей, бизнес-случаев, объектов и таблиц. Эти сущности могут создаваться, открываться, редактироваться или перемещаться в различные группы (папки). В проводнике ARIS Toolset различные БД могут обмениваться содержанием, используя технологию “drag and drop”.

Самым высоким уровнем организации модели являются локальные серверы или сети серверов, работающие в многопользовательской среде. БД могут создаваться, открываться, сниматься с регистрации и удаляться на локальных серверах. БД открывается путем ее вызова и введения ID пользователя. Атрибуты БД могут редактироваться путем двойного нажатия на них кнопки мыши.

Структура БД определяется группами (папками), в которые при сохранении загружаются модели и объекты, а также бизнес-случаи и таблицы. Для организации структуры БД важную роль играет организация проекта, поскольку привилегии доступа к содержанию БД контролируются через группы моделей.

Дизайнер используется для создания и редактирования моделей. Объекты, связи и другая информация по этим моделям хранится в БД и может использоваться в различных отчетах, при анализе БП и для многих других целей. Дизайнер проверяет новые объекты во время моделирования на предмет их возможного существования в БД. Если объекты с введенными названиями уже существуют в БД, то выводится их перечень и пользователю предоставляется возможность их повторного использования. Модели могут быть представлены в диаграммном или текстовом виде. Полноэкранный режим и навигационные возможности обеспечивают удобный обзор и поиск моделей.

Таблицы используются для хранения данных, а их интерфейс базируется на табличных процессорах. Все объекты и их отношения снабжены атрибутами, которые могут редактироваться в таблицах. Процесс редактирования подобен работе со стандартными табличными процессорами. Таблицы используются не только для редактирования атрибутов, но и для вывода результатов имитационного моделирования в ARIS Simulation и введения данных для функционально-стоимостного анализа в процессе расчетов в дополнительном модуле ARIS ABC. Значения атрибутов могут наследоваться из других приложений.

В ARIS используется технология Мастеров (Wizards) для управления последовательностью операций. Мастера проводят пользователя в интерактивном режиме через определенную последовательность

действий. Они также позволяют вернуться к предыдущим шагам. Технология мастеров используется, например, для создания БД, моделей, генерации отчета, для расчетов по ABC-методологии и т. д.

Все продукты ARIS являются клиент-серверными приложениями и могут работать в многопользовательской среде. Для расширения информационной насыщенности моделей с ее объектами могут быть связаны рисунки, текстовые документы, презентации, видеозаписи, Web-страницы и многое другое. ARIS-модели можно дополнить рисунками и текстами. Существует два способа интеграции моделей с перечисленными сущностями. Один – присвоить их атрибуту объекта, другой – установить связь с загруженными файлами через технологию OLE или напрямую соединить их в модели. Присвоенные документы, программы и адреса могут быть вызваны из модели и, таким образом, дополнять ее описание. Эти присвоения имеют особое значение в области управления знаниями, при проектировании ИС, поддерживающих систему управления качеством, при разработке процессно-ориентированных курсов обучения и для других целей. Например, документы, относящиеся к системе качества и хранящиеся в текстовом процессоре “Word”, могут вызываться и редактироваться прямо из моделей.

Весомость практической информации для лица, которому она предназначена, зависит от доверия к ее автору (его авторитета, репутации у специалистов и т. д.). По этой причине в ARIS указывается автор каждого объекта, время его создания и многое другое. Вся эта информация хранится в едином репозитории, что обеспечивает целостность и непротиворечивость данных, а также эффективность процессов моделирования и анализа. ARIS использует объектноориентированную БД “POET”.

В стандартный пакет ARIS Toolset, ARIS Easy Design и ARIS for mySAP.com включены *фильтры методов (method filters)*. Они сжимают всю совокупность моделей до необходимого уровня, что позволяет облегчить процесс моделирования. Фильтры методов, определенные пользователем или системным администратором, ограничивают количество объектов и значительно уменьшают их сложность. В среде ARIS Toolset можно создавать новые фильтры методов или модифицировать уже существующие. Это делается на предварительной проектной стадии, когда определяются соглашения моделирования и формируется соответствующий фильтр методов для отдельных работников. Конфигурация фильтров методов осуществляется в “Конфигураторе” Explorer, который позволяет создавать, редактировать и документировать фильтры методов



для эффективной компоновки типов моделей, типов атрибутов моделей, типов объектов, типов атрибутов объектов, типов связей, типов атрибутов связей и т. д.

На рис. 6.1.3 представлены все типы моделей, которые допустимы в ARIS 5.0 с тремя из следующих восьми фильтров.

1. ARIS for MySAP.com
2. ARIS Simulation and ARIS ABC Filter
3. Balanced Scorecard Filter
4. Easy Filter
5. Entier Method (Fullmethod)
6. Extended Standard Filter
7. SAP Filter
8. Standard Filter

В Приложении 1 приведена таблица, в которой для всех видов моделирования перечислены названия (на английском и русском языке) большинства типов моделей с краткими описаниями их назначения и перечнем номеров фильтров, в которых допустимо их использование.

В таком множестве моделей можно затеряться. Поэтому все ARIS-модели можно разбить на классы, группы, виды, уровни и типы. Различают следующие классы моделей.

**Строгая иерархия.** Модель такого класса, например, дерево функций, включает в себя только один тип объекта, а иерархические связи могут быть организованы с помощью нескольких типов соединения.

**Строгая иерархия с окружением.** Модель такого класса, например, диаграмма целей, представляет собой дерево, которое строится на основе одного типа объекта, а иерархические связи могут быть организованы с помощью нескольких типов соединения. С объектами, составляющими иерархию, могут быть связаны другие типы объектов.

**Иерархия с перекрестными соединениями.** Модель такого класса, например, диаграмма структуры знаний, представляет собой иерархию, которая включает различные типы объектов, где иерархические связи могут быть организованы с помощью нескольких типов соединения. В отличие от моделей со строгой иерархией объектов, которые входят в иерархию с перекрестными соединениями, могут быть связаны друг с другом как напрямую, так и через другие типы объектов.

**Тип основного объекта.** Модель такого класса, например, диаграмма окружения функций, отображает связь между одним типом объекта и другими типами объектов. Все типы объектов, прикрепленные к типу основного объекта, имеют одинаковые приоритеты.

**Приоритетный тип основного объекта.** В модели с приоритетным типом основного объекта отобража-

ются связи между типом основного объекта и другими типами объектов, где тип объекта, имеющий наивысший приоритет, играет основную роль в иерархии.

**Направленный граф с окружением.** Модель такого класса, например, диаграмма eEPC, содержит особые типы объектов, которые формируют направленную структуру и могут быть связаны с другими типами объектов.

**Ненаправленный граф с окружением.** Модель такого класса, например, диаграмма в виде столбцов или строк, содержит структурно-зависимые типы объектов, соединенных связями без стрелки.

ARIS-модели можно разбить на три группы. К первой относятся *основные модели*, например, модель цепочки, добавляющей качество, организационная схема, дерево функций, модели eEPC, eERM, PCD и другие. Во вторую группу входят *расширения основных моделей*. К ним относятся карта знаний, диаграмма структуры знаний, диаграмма окружения функции и другие. Модели третьей группы – это модели, *ориентированные на отдельные области моделирования*, например, внедрение системы SAP R/3, документация плана развития, объектно-ориентированное моделирование, описание потока материалов, моделирование workflow, функционально-стоимостный анализ, управление проектом и т. д.

В заключение перечислим функциональные возможности основных и вспомогательных ARIS-модулей.

*ARIS Easy Design* позволяет выполнять следующие операции: администрировать БД (*ARIS Administrator*), управлять атрибутами ARIS-сущностей (*ARIS Attributes*), контролировать изменения атрибутов (*ARIS Change Managements*), импортировать методологические фильтры и шаблоны из одной БД в другую (*ARIS Configuration*), конструировать модели (*ARIS Designer*), работать с ARIS-сущностями (*ARIS Explorer*), экспортировать или импортировать БД в формате ASCII (*ARIS Export/Import*), автоматически идентифицировать ARIS-сущности без управления префиксами (*ARIS Identifier*), осуществлять слияние моделей и объектов (*ARIS Merge*), генерировать отчеты (*ARIS Report*), редактировать текстовые документы (*ARIS RTF Editor*), выполнять семантические проверки моделей (*ARIS Semantic Check*), отображать варианты моделей (*ARIS Variants*). Здесь в скобках указаны программные средства (модули), которые обеспечивают выполнение перечисленных операций.

Наряду с перечисленными операциями *ARIS Toolset* позволяет анализировать модели и осуществлять их анимацию (*ARIS Analysis*), создавать и использовать графические диаграммы (*ARIS Chart*), объединять БД моделей (*ARIS Consolidation*), осуществлять слияние объектов и моделей, включая одинаковые глобальные идентифика-

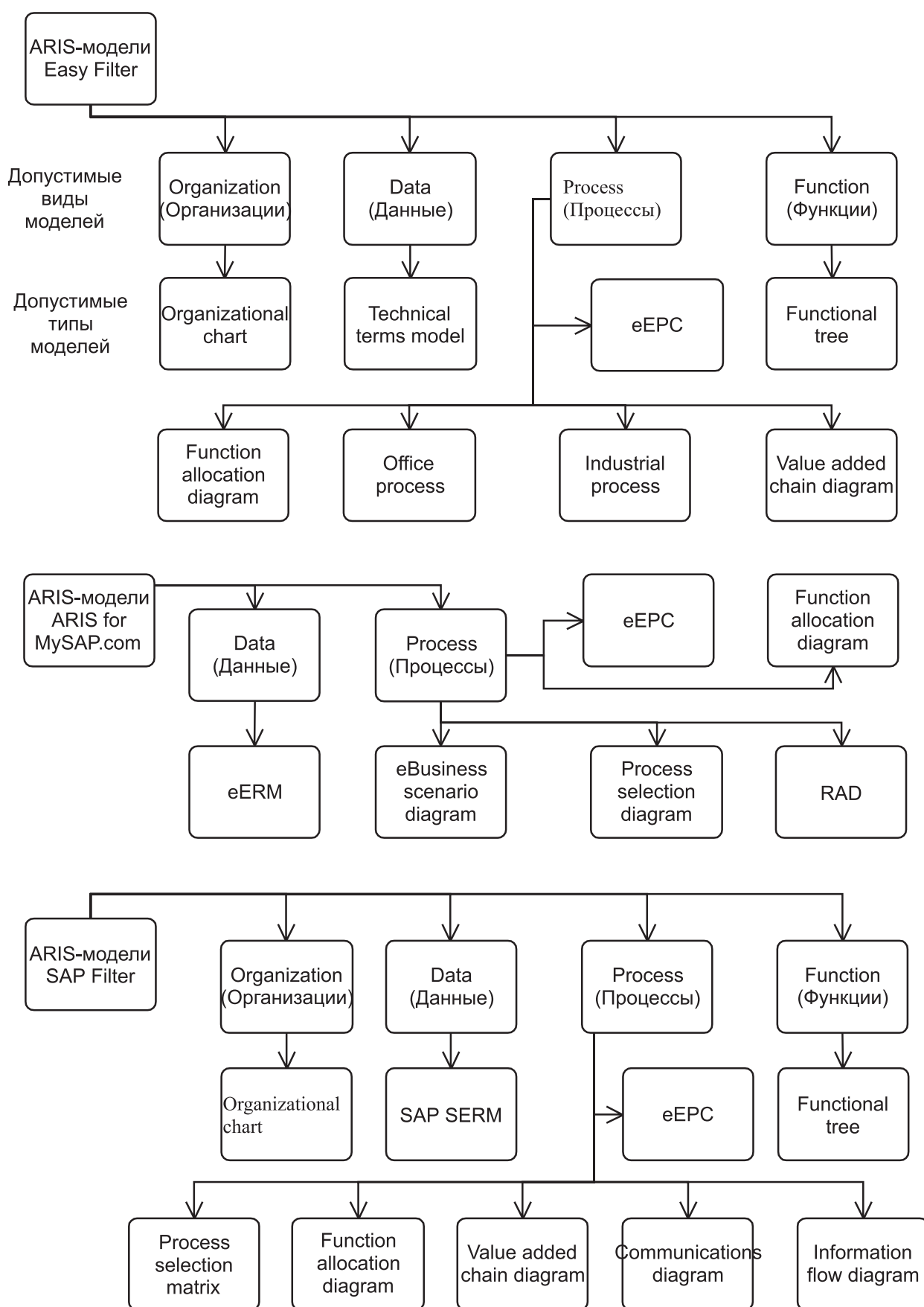


Рис.6.1.3 Примеры Структурных моделей (Structuring Model) возможных типов ARIS-моделей с тремя различными фильтрами.



торы (*ARIS Merge*), создавать новые модели на основе уже существующих (*ARIS Model Generator*), осуществлять семантические проверки моделей (*ARIS Semantic Check*), создавать скрипты отчетов (*ARIS Script Editor*), имитировать выполнение процессов (*ARIS Simulation*), создавать варианты моделей, удалять связи между вариантами, выполнять операции сравнения вариантов (*ARIS Variants*), генерировать новые объекты и модели в программе Excel и переносить их в ARIS (*ARIS Process Generator*). В ARIS имеется возможность с помощью специального отчета перенести объекты и модели в Excel и вернуть измененные данные назад в ARIS.

Дополнительные ARIS-модули, требующие дополнительных лицензий, предназначены для решения некоторых частных задач. *ARIS ABC* выполняет функционально-стоимостный анализ моделей. *ARIS Connectivity for Lotus Notes* позволяет 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* переносит ARIS-модели 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* служит для обмена информацией БД моделей ARIS с программными приложениями других производителей (с так называемыми приложениями партнеров). *ARIS Web Publisher* служит 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* используется для стратегического управления. *ARIS API (Application Programming Interface)* является интерфейсом между архивом данных ARIS и пользовательским приложением, который 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++. Эта библиотека разработана как динамически подключаемая библиотека, чтобы исключить зависимость от специфики конкретного компилятора языка программирования. Поэтому разработчик может выбрать удобную для себя программную среду. Модуль *ARIS API* предоставляет доступ к данным, созданным с помощью системы ARIS. Поэтому эти данные можно использовать в других приложениях или дополнять существующие данные в архиве ARIS

новыми объектами. При помощи *ARIS API* можно перенести из архива или в архив результаты моделирования в системе ARIS, создавать свои надстройки к системе ARIS, интегрировать информацию в едином стандарте ARIS.

Каждый из перечисленных модулей 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.

## 6.2. Основные свойства моделей и объектов

Перед построением ARIS-диаграмм необходимо выполнить глобальные и локальные настройки среды моделирования под свои требования. Для этого используются различные возможности (главное меню, выпадающие подменю, всплывающие меню, панели инструментов и т. д.). Панели инструментов предоставляют возможность пользователю выполнять те или иные действия, выбирая желаемый пункт меню или нажимая кнопкой мыши на пиктограмму, соответствующую данному пункту меню. Например, если находясь в окне моделирования нажать на кнопку *Full Screen* (полный экран) в подменю пункта View главного меню, то активизируется режим полного экрана. При этом граница окна, зона меню, панель инструментов и другие составляющие интерфейса исчезнут, предоставляя пользователю максимально возможную площадь для просмотра модели. Для того, чтобы выйти из режима полного экрана, достаточно нажать Esc.

На рис.6.2.1 изображен вид ARIS-экрана, где представлено большинство из следующих допустимых панелей инструментов: *Animation Control* (Управление анимацией), *Attributes* (Атрибуты), *ARIS Explorer* (Проводник ARIS), *Chart* (Диаграмма); *Debug* (Отладчик), *Explorer* (Проводник), *Insert* (Вставка), *File* (Файл), *Formatting* (Формат), *Macros* (Макрос), *Model View* (Вид моделей), *Modeling* (Моделирование), *Script Layout* (Макет скриптов), *RTF Editor* (Текстовый редактор).

На рис.6.2.1 к основному окну моделирования 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 представлена модель в уменьшенном масштабе. Белый прямоугольник охватывает часть модели, которая отражается в основном окне. Передвигая этот прямоугольник кур-

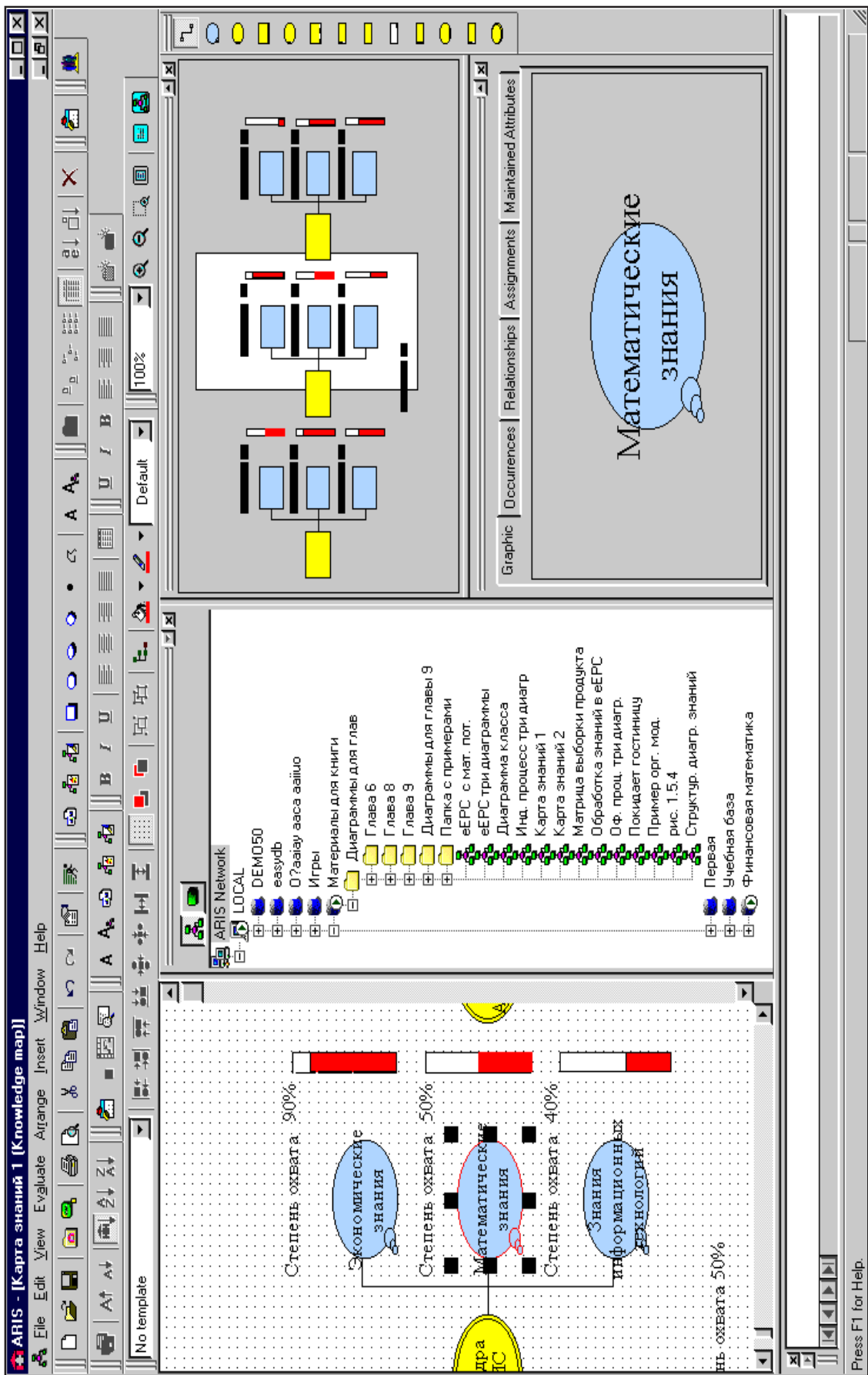


Рис.6.2.1 Окно ARIS Easy Design с набором панелей инструментов.

сором мыши, можно изменять представление в основном окне. *Окно просмотра объекта* (команда меню *View/Object Window*). Это вспомогательное окно для работы с выбранным объектом модели. Оно используется для просмотра его связей и т.п. *Окно вывода информации* (команда меню *View/Output Window*). Это вспомогательное окно для вывода системной информации для пользователя. *Окно проводника* (команда меню *View/Explorer Window*). Это окно *ARIS Explorer* предназначено для работы с ARIS-сущностями.

Описанию возможностей настройки ARIS-сущностей (серверов, БД, групп, языков, инструментов и т. д.) и работы с элементами управления в окнах на рис. 6.2.1 можно посвятить отдельную книгу. Поэтому здесь рассматриваются только основы настройки свойств моделей, а также их атрибуты и основные атрибуты объектов.

Каждая модель обладает определенными свойствами. Некоторые из них задаются автоматически системой ARIS при создании модели, другие задаются пользователем самостоятельно.

С моделями можно осуществлять следующие действия: проверять корректность (семантические проверки), составлять разнообразные отчеты, непрерывно улучшать, оптимизировать по различным критериям, анализировать по различным методикам (например, функционально-стоимостный анализ), сравнивать друг с другом, конвертировать в другие программные системы. Перечисленные действия обеспечиваются при помощи следующих операций: копирования моделей, создания вариантов, слияния моделей, экспортирования и импортирования моделей, хранения моделей, резервирования копий, консолидации моделей.

Модель состоит из набора объектов, каждый из которых имеет уникальное имя, принадлежит к определенному типу объектов, соединен одной или несколькими связями с другими объектами, имеет определенное значение в модели, описывается конкретными свойствами, может использоваться в других типах моделей, может иметь свои экземпляры, представленные одним или несколькими символами. Внешний вид экземпляра объекта может быть изменен в определенных пределах. В приложении 2 приведены изображения ARIS-объектов и даны их краткие описания.

Объекты бывают *структурно-независимыми* и *структурно-зависимыми*. Последние определяют общую структуру модели.

Объекты можно описывать более подробно с помощью детализирующих моделей. Для этого используется связь типа *детализации*, которая неявно проводится от типа объекта (базового объекта) к типу детализи-

рующей модели. Экземпляр такой связи, т. е. связь между определенным объектом и определенной моделью, называется *детализацией*. Допустимые типы детализации зависят от типа базового объекта и модели, в которой он используется.

Важную роль при моделировании играет правильный выбор связей (отношений) между объектами в модели. Связь описывает взаимоотношения между двумя объектами и имеет определенные свойства. Каждая связь имеет уникальное имя и принадлежит к определенному типу. *Тип связи* зависит как от типов соединяемых объектов, так и от направления соединения. Различают объекты-источники связи и объекты-приемники связи. Между двумя объектами может быть создана одна или несколько связей различных типов. Между некоторыми типами объектов связи не могут существовать. В приложении 3 для каждого из восьми типов моделей в ARIS 5.0 с простым фильтром перечислены типы связей (отношений), проводимых между объектами.

У связи могут быть свои экземпляры. Внешний вид экземпляров связи может быть изменен в определенных пределах.

Среди связей различают *соединения* и *детализирующие связи*. Экземпляры соединений представляют собой видимые прямые или ломаные линии 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.

Каждая ARIS-сущность (модель, объект, связь и т. д.) обладает определенными свойствами, которые должны быть описаны. Одной из составляющих свойств являются атрибуты. *Атрибут* – необходимое, существенное, неотъемлемое свойство сущности. Естественно, что разные сущности имеют разные атрибуты.

Установка свойств модели происходит в диалоговом окне (ДО) свойств модели, которое можно вызвать несколькими способами. Например, нажать на правую кнопку мыши, когда ее указатель находится в поле моделирования, и в появившемся всплывающем меню выбрать пункт *Properties*, либо нажать на панели инструментов кнопку с тем же именем. Внешний вид ДО свойств модели показан на рис. 6.2.2. Это ДО содержит вкладки *Model Appearance* (изображение модели), *Attributes* (атрибуты), *Maintained Attributes* (поддерживаемые атрибуты) и другие. Рассмотрим содержание перечисленных вкладок.

## Model Appearance

Вкладка “Изображение модели” позволяет менять

параметры, связанные с внешним видом диаграммы. Она предоставляет следующие возможности: выбор цвета фона, использование выбранного фона во всех моделях, изменение масштаба печати, выбор способа расположения текста внутри символа объекта, расположение новых связей, удаление цвета за текстом, выбор типа изображения модели, использование сетки и др.

По умолчанию в ARIS задан белый цвет фона области для построения диаграмм. Пользователь может изменить цвет области модели, используя раскрывающийся список цвета фона. При раскрытии списка появляется палитра доступных цветов. После выбора цвета из этой палитры цвет фона области для моделирования изменится на желаемый. Выбранный таким образом фон используется только в данной модели.

Меняя размер страницы (шкалу распечатки), можно модифицировать размер модели для вывода на печать. При этом размер диаграммы в окне моделирования не изменится. Открывая окно предварительного просмотра распечатки, введенная величина автоматически передается в текстовое поле выбора шкалы распечатки. При закрытии модели установленная величина сохраняется.

Способ расположения текста, записываемого внутри символа (пиктограммы, картинки) изображаемого объекта, определяет, каким образом этот текст связан с размерами картинки изображения объекта. Существуют следующие способы расположения текста:

а) произвольное, т. е. текст может выходить за границы объекта;

б) с изменением размеров объекта, т. е. размеры объекта увеличиваются до размеров текста. Если размер текста уменьшается, то размер объекта также уменьшается, но до тех пор, пока он не достигнет стандартного размера. При применении этого способа отсутствует возможность масштабирования объектов;

в) урезание текста, т. е. текст, выходящий за границы объекта, становится невидимым, так как он урезается до размеров объекта;

г) сокращение текста до одной строки, т. е. часть текста, не вмещаемая в одну строку, становится невидимой и заменяется троеточием.

Для того, чтобы текущая модель печаталась как черно-белая, необходимо выбрать флажок Black and White. Если модель должна печататься в цвете, но используется черно-белый принтер, то цвета будут выводиться с различной градацией яркости (как оттенки серого цвета). Линии связи и строки текста также отобразятся как черно-белые. Цвет изображения можно задать и при предварительном просмотре перед печатью.

Выбор сетки помогает при размещении, перемеще-

нии и выравнивании объектов, а также при установлении связей между ними. Если переключатель использования сетки активизирован, то точечная разметка сетки устанавливается в пределах выбранной ширины. Расстояние между точками сетки регулируется путем перемещения движка ширины сетки. Эта величина лежит в пределах от 2 до 20 условных единиц. В нижней части ДО указывается тип текущей модели и путь к группе (папке), в которой она сохранена.

## Attributes

Атрибуты позволяют записывать и модифицировать характеристики моделей. Они формируют содержимое БД. Атрибуты представлены в табличной форме, что обеспечивает их быстрый обзор и позволяет сравнивать значения атрибутов, размещенных в смежных столбцах. Например, можно просмотреть значения атрибутов сразу нескольких объектов и сравнить их. Автоматическая группировка атрибутов позволяет осуществить быстрый доступ к желаемой информации и облегчает их редактирование даже при использовании больших массивов данных. Значения атрибутов могут быть полностью или частично скопированы и затем включены в другие приложения. Конкретное значение атрибута может быть заменено данными из другого приложения, например, из Word или Excel.

Рассмотрим основные атрибуты ARIS-сущностей, т. е. объектов, моделей, групп и т. д.

*Атрибут Name (Имя)* позволяет давать имя сущности. Он является стандартным типом атрибута. По умолчанию в графическом символе объекта появляется название атрибута имени. Этот атрибут часто используется при выборе объектов из списков (например, создавая копию, при навигации, консолидации объектов). Большое значение имеет исключительность имен. ARIS помогает поддерживать исключительность имен при создании новых объектов. Имя и его исключительность особенно важны при объединении БД, при этом атрибут (как альтернатива идентификатора) может использоваться для того, чтобы однозначно идентифицировать объекты.

*Атрибут Identifier (Идентификатор)* содержит уникальное имя сущности (например, 21.77, ABC.31). Он является стандартным типом атрибута. Этот атрибут позволяет автоматически назначать идентификаторы. При этом он состоит из префикса, который может определяться свободно, и числового индекса, который подсчитывается автоматически. Этот атрибут, главным образом, используется в двух действиях ARIS. Во-первых, при выборе объектов для создания их копий (нажатие правой кнопки мыши при позиционировании объекта). Во-вторых, при



объединении, когда можно использовать имя и идентификатор для распознавания одинаковых объектов.

*Атрибут Full name (Полное имя)* является стандартным атрибутом, который присутствует во всех моделях и объектах. С помощью этого атрибута задается полное имя для соответствующей сущности.

*Атрибут Type (Тип)* является стандартным атрибутом, который присутствует во всех моделях и объектах. Он задается по умолчанию и не может быть изменен.

*Атрибут Description/Definition (Описание/Определение)* служит для более полного текстового описания сущности.

*Атрибут Remark/Example (Замечание/Пример)* служит для более подробного описания сущности посредством примеров, что облегчает пользователю понимание этой сущности (например процесса). Этот атрибут может содержать до 3000 символов, т. е. целую страницу текста.

*Атрибут Connection role (Роль связи)* используется для подробной спецификации связей в модели, т. е. через него можно дать более точное описание связей между двумя объектами с помощью самостоятельно вводимого текста.

*Атрибут Link (Связь)* может использоваться для запуска на выполнение программы с объекта, модели и т. д. В нем содержится путь к файлу и/или прикладной системе, которые можно запустить на выполнение (например, C:\winword\winword.exe sample.doc).

*Атрибут Avg. orientation time (Среднее ориентировочное время)* определяет среднее время, необходимое для подготовки субъекта к выполнению функции.

*Атрибут Avg. processing time (Среднее время процесса)* указывает объем времени, которое необходимо для выполнения функции.

Кроме вышеперечисленных, ARIS-сущности могут иметь следующие атрибуты: создатель (Creator), номер факса (Fax number), адрес электронной почты (E-mail address), адрес (Address), телефонный номер (Telephone number), параметр (Parameter), средняя стоимость материалов (Avg. material costs), средние прочие затраты (Avg. other costs), средние затраты на персонал (Avg. Personnel costs), средние общие затраты (Avg. total costs), среднее время ожидания (Avg. wait time), количество служащих (Number of employees) и другие.

Атрибуты, отражающие материальные затраты, находятся в группе затрат. Они выражаются значением валюты и числа, округленного до двух десятичных знаков. Временные характеристики процесса содержатся в группе Time (время) и могут выражаться в минутах, часах, днях и т. д. Атрибуты материальных и временных затрат принадлежат объекту типа *функция*. Такие атрибуты, как адрес, телефонный номер, факс, адрес

электронной почты, принадлежат объектам типа *персона*. Атрибут, содержащий информацию о количестве служащих, принадлежит объекту типа *организационная единица*. Роль связи указывается для различных связей между объектами и т. д.

Внешний вид вкладки атрибутов представлен на рис.6.2.3. В левой области ДО атрибутов находится дерево с названиями возможных групп атрибутов. Перемещение по этому дереву аналогично движению в проводнике. Открыть атрибуты, входящие в ту или иную группу, можно двойным нажатием левой кнопки мыши на имени группы, либо единственным нажатием кнопки плюс. При этом в правой части ДО появляются названия атрибутов выбранной группы. Список всех доступных атрибутов можно открыть, нажав правую кнопку мыши и выбрав в появившемся всплывающем меню пункт Select all (выбрать все). Закрыть список атрибутов можно повторным двойным нажатием на имени группы или единственным нажатием на кнопку минус. Список всех доступных атрибутов можно закрыть, нажав правую кнопку мыши и выбрав в появившемся всплывающем меню пункт Close all (закрыть все).

Области ДО разделяются перемещаемой полосой. Для того, чтобы изменить размер областей необходимо переместить указатель мыши к полосе, нажать на правую кнопку мыши и передвинуть разделительную полосу. В правой части окна находится таблица, первая колонка которой содержит названия атрибутов. Они выделены серым цветом и не могут быть изменены пользователем. Второй столбец содержит строки для ввода конкретных значений атрибута. Значения большинства атрибутов определяются пользователем и могут быть изменены им в любой момент.

## Maintained Attributes

Внешний вид вкладки поддерживаемых атрибутов показан на рис.6.2.4. Эта вкладка показывает значения атрибутов и в ее нижней части можно увидеть глобальный уникальный идентификатор (Guid). Он представляет сочетание различных знаков (цифр, символов, тире).

ARIS-сущности имеют международный уникальный идентификатор, причем БД, таблицы, модели, объекты и связи приобретают Guid при создании. Если создаются варианты этих сущностей, то они содержат ссылки на мастера Guid. После того, как варианты отредактированы в различных БД, глобальный идентификатор гарантирует, что ссылки восстановятся при слиянии.

Таким образом, каждая модель, создаваемая пользователем, имеет ряд свойств. Установка этих свойств происходит в ДО свойств модели. Свойства, являясь

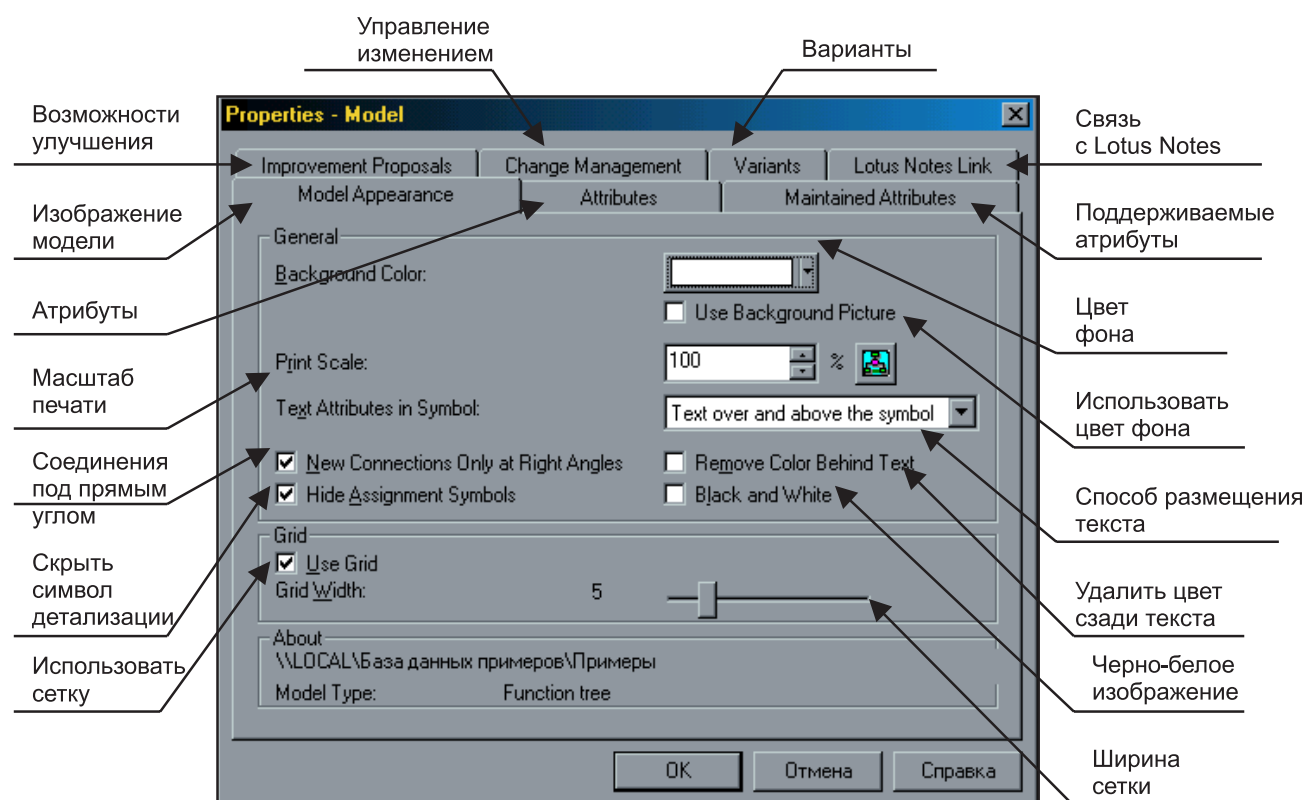


Рис.6.2.2 Внешний вид диалогового окна свойств модели и вкладки изображения.

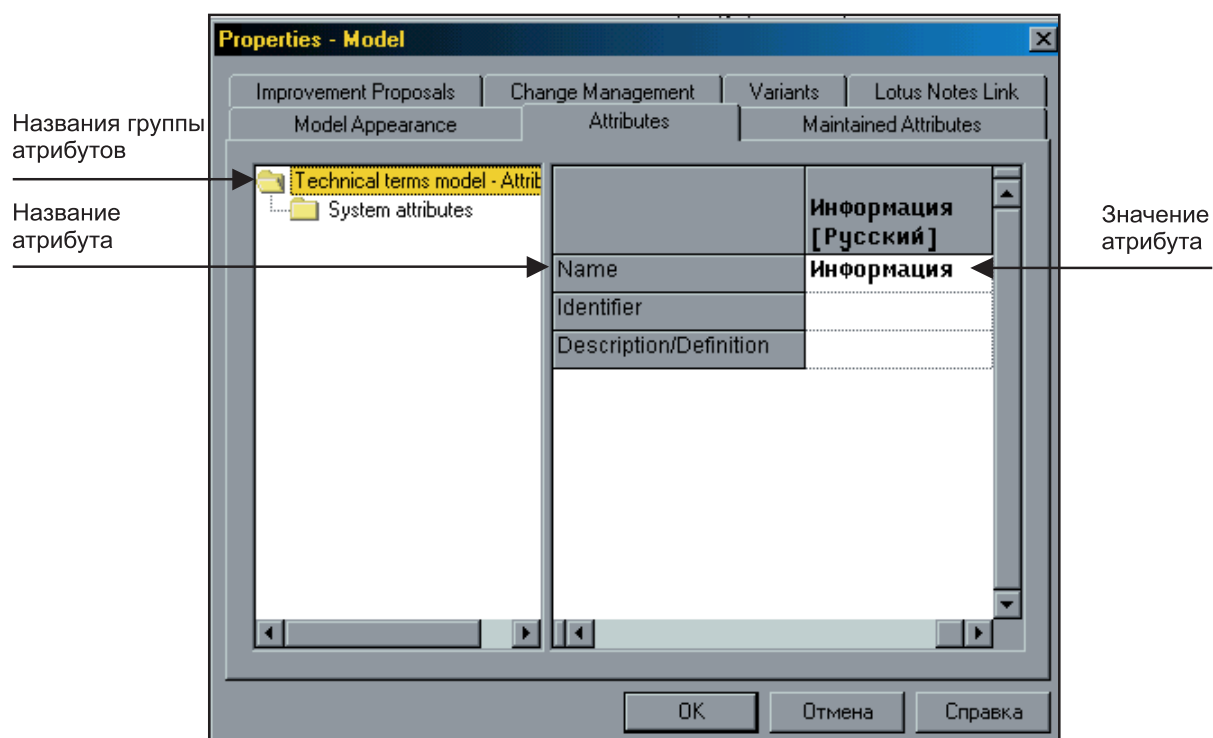


Рис. 6.2.3 Внешний вид вкладки атрибутов.



внутренней информацией сущности, связаны либо с внешним видом модели или объектов модели, либо содержат информацию о моделируемой системе.

Рассмотрим свойства объектов. Каждый объект в системе ARIS обладает определенными свойствами. Их установка происходит в *ДО свойств объекта*, содержащем ряд вкладок, аналогичных вышеописанным вкладкам *ДО свойств модели*. Кроме того, *ДО свойств объекта* содержит ряд дополнительных вкладок: связи (Relationships), расположение атрибута (Attribute Placements), назначенные модели (Assignments), изображение объекта (Object Appearance) и другие. Рассмотрим содержание перечисленных вкладок.

## Relationships

Внешний вид вкладки связей изображен на рис.6.2.5. Эта вкладка отображает все связи объекта в БД. Как только выбирается вкладка связей, программа автоматически находит в БД связи выбранного объекта. Для всех объектов, с которыми связан выбранный объект, указывается имя и группа.

Вкладка связей позволяет также просмотреть и отредактировать свойства модели, в которой находится указанный объект. Для этого необходимо нажать на имя желаемого объекта правой кнопкой мыши. Тогда появится всплывающее меню, содержащее единственный пункт Properties. При выборе данного пункта появляется *ДО свойств объекта*, содержащее имена моделей (рис.6.2.6). Если имя модели не выбрано, кнопки Open Model и Properties не доступны. При выборе имени модели эти кнопки активизируются, выбрав одну из них, можно перейти к редактированию свойств модели.

## Attribute Placements

Внешний вид вкладки расположения атрибутов изображен на рис.6.2.7. Эта вкладка позволяет расположить значение атрибута выбранного объекта в желаемой позиции. Список атрибутов позволяет выбрать их имена. Он включает все атрибуты объекта, которые доступны в используемом фильтре. Эти атрибуты размещены в алфавитном порядке. При выборе атрибута из списка его имя появляется в текстовом поле Selected Attribute. Атрибуты могут устанавливаться на объекте в одиннадцати позициях, которые представлены квадратами в поле размещения.

Атрибут имени обычно устанавливается в центре при создании нового объекта. Размещение других атрибутов производится пользователем. Для выбора конкретного расположения атрибута необходимо активизировать соответствующий квадратик. Вкладка располо-

жения атрибута позволяет производить выравнивание текста (значения) атрибута по определенному критерию. Если расположение атрибута выбрано, то предлагается три кнопки выбора для выравнивания текста: по левому краю, по правому краю, по центру. В раскрывающемся списке доступных форматов шрифта пользователю предлагается выбрать формат шрифта. Эта возможность предлагается при условии, что месторасположение атрибута установлено. Некоторые атрибуты могут быть отображены в виде иконки. Кроме того, предлагается отобразить имя соответствующего атрибута дополнительно к значению данного атрибута. При активизации переключателя With Attribute Name имя атрибута появится перед его значением.

Для удаления всей введенной информации о размещении атрибутов достаточно нажать на кнопку Remove All Placements. Для изменения выбранного месторасположения атрибута нужно убрать флажок в соответствующем квадрате в поле размещения и нажать в другом квадрате для нового размещения атрибута. При этом выбранный атрибут устанавливается в выбранной новой позиции. Для сохранения сделанных изменений необходимо закрыть *ДО свойств* нажатием кнопки ОК.

## Assignments

Внешний вид вкладки назначений аналогичен *ДО*, изображенному на рис.6.2.8. В этом *ДО* имеются две дополнительные кнопки: New (новая) и Delete (удалить). Вкладка назначений включает все модели, которые можно использовать для детализации текущего объекта. Как только выбрана вкладка назначений, программа ищет назначения для выбранного объекта в БД. Для всех моделей БД, которые назначаются выбранному объекту, указывается имя и группа.

## Object Appearance

Внешний вид вкладки изображения объекта показан на рис.6.2.9. Используя возможности, предоставляемые данной вкладкой, можно изменить внешний вид объектов. Выбранным объектам можно назначить стандартный вид путем активизации встроенного переключателя. При этом определенные пользователем значения блокируются.

При выборе переключателя Shaded (затенение) графический символ объекта изображается с затенением. Цвет и атрибуты элементов изображения для объектов офисных и промышленных процессов модифицировать нельзя. Если встроенный переключатель стандартных величин не активизирован, то пользователю предоставляется возможность выбора цвета объекта, изменения цвета линии границы объекта, ее стиля

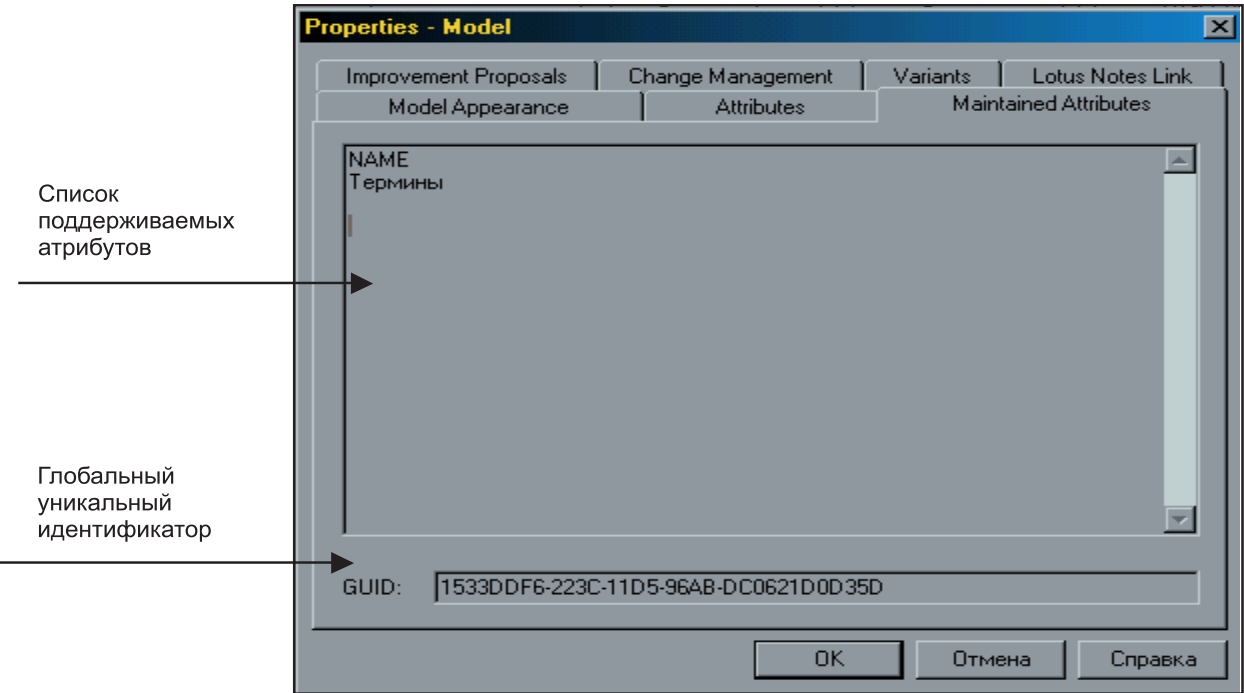


Рис. 6.2.4 Внешний вид вкладки поддерживаемых атрибутов.

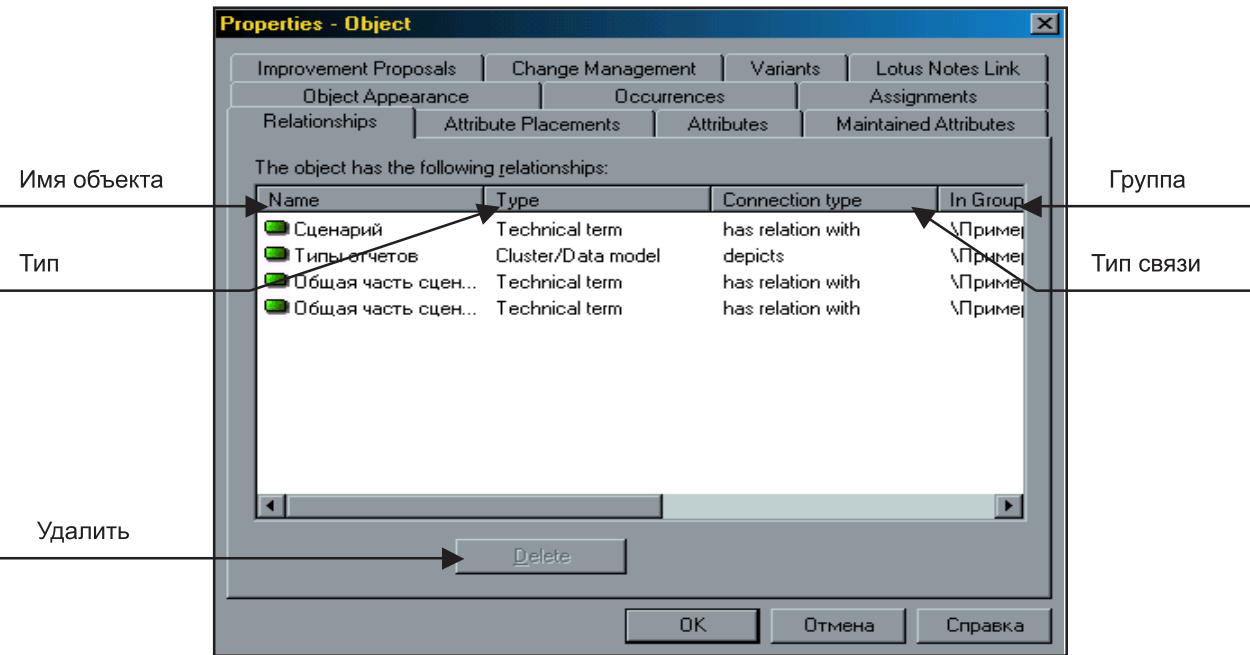


Рис. 6.2.5 Внешний вид вкладки связей.

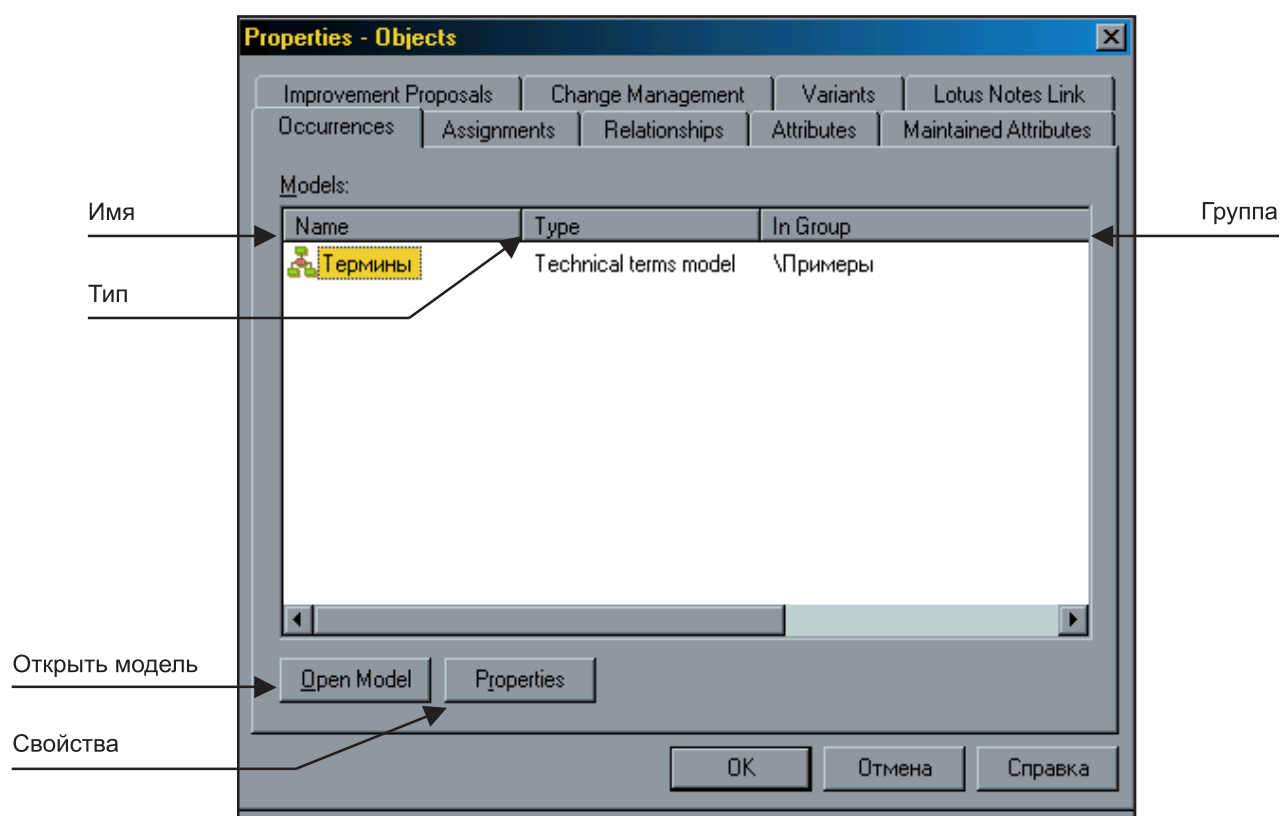


Рис. 6.2.6 Внешний вид ДО свойств объекта.

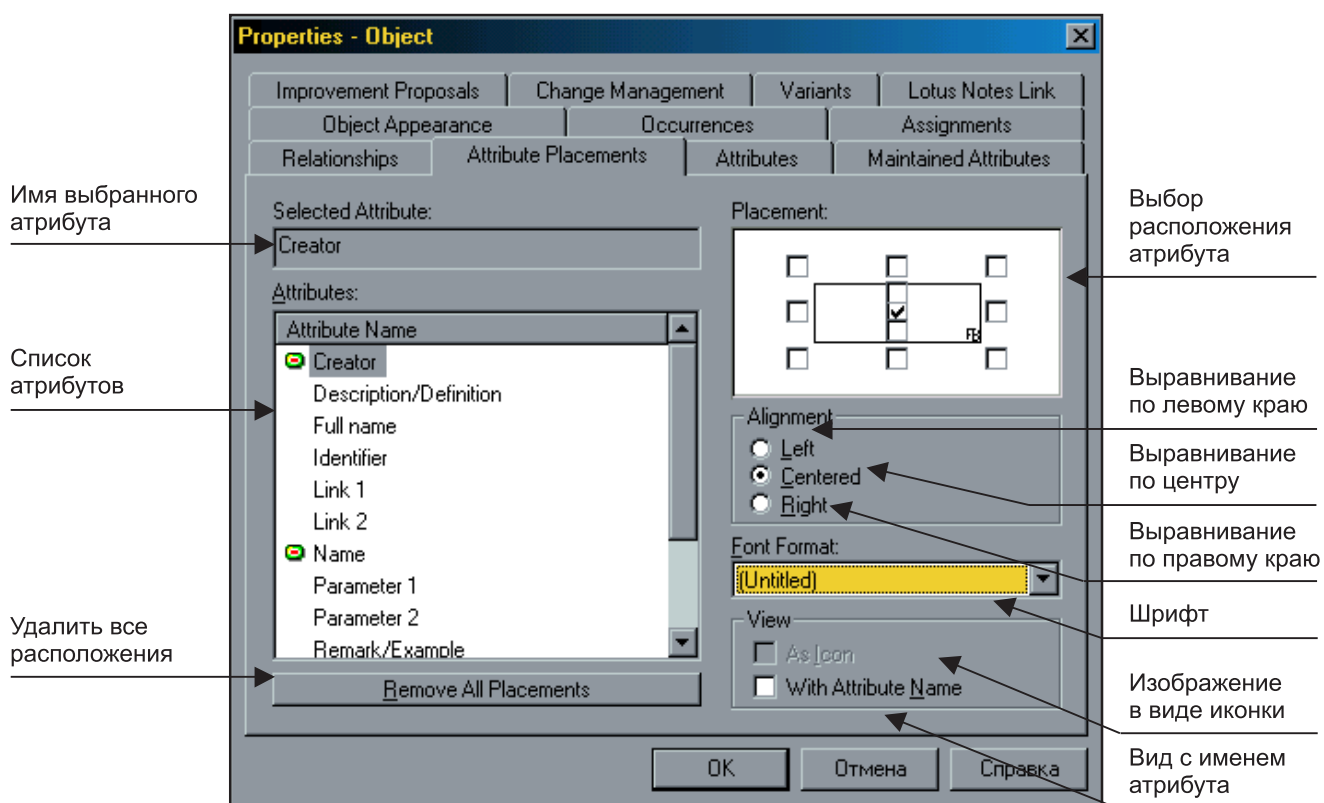


Рис. 6.2.7 Внешний вид вкладки расположения атрибутов.

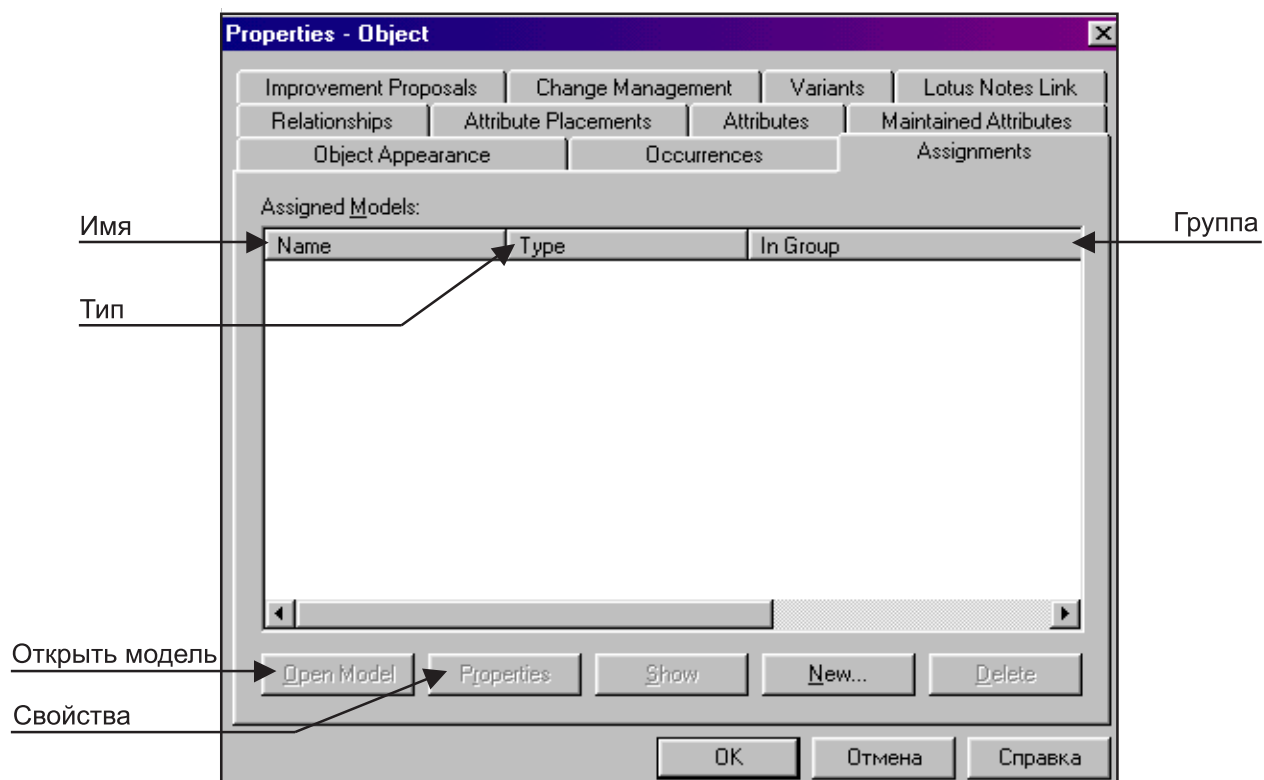


Рис. 6.2.8 Внешний вид вкладки назначений

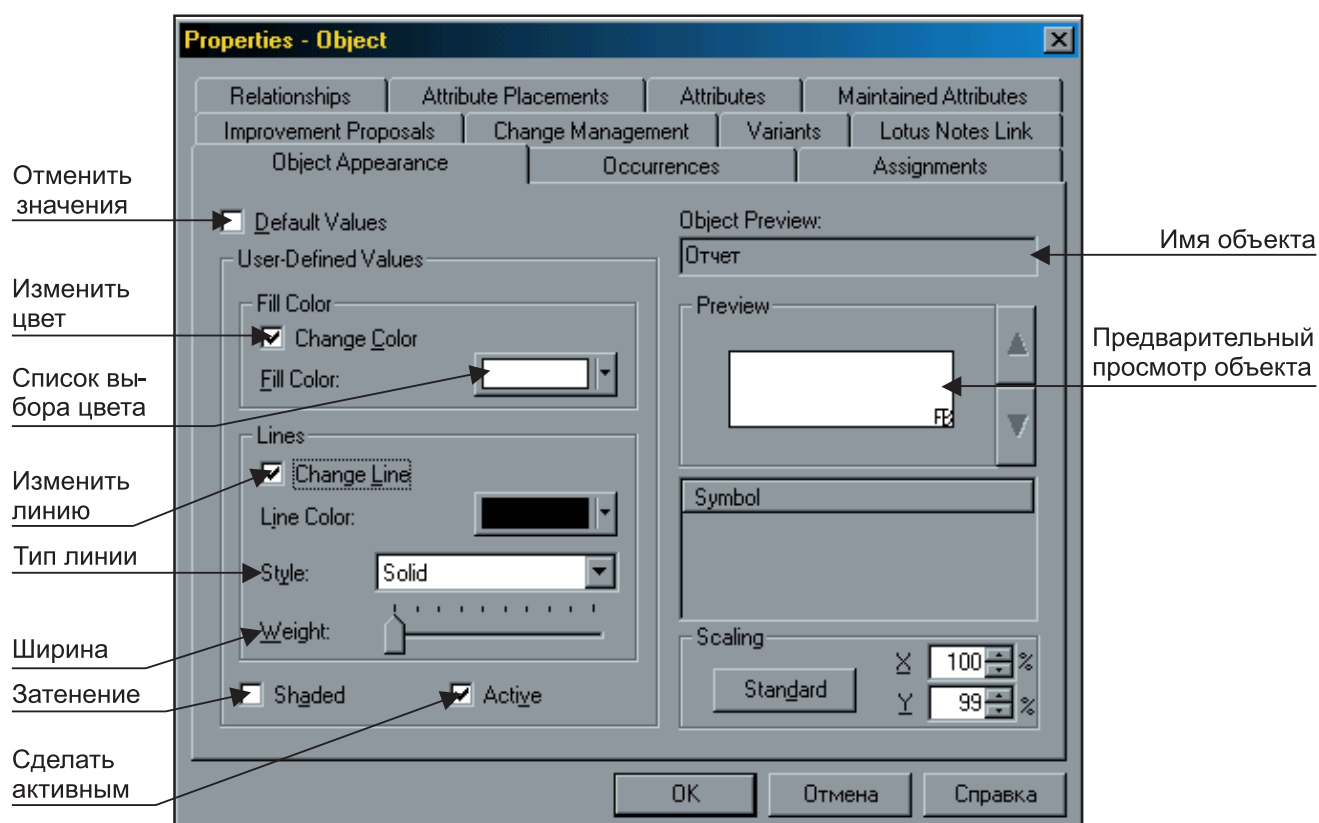


Рис. 6.2.9 Внешний вид вкладки изображения объекта

и толщины. Цвет объекта выбирается из предлагаемой палитры цветов. Для выбора дополнительных цветов достаточно нажать на кнопку Other. Для границ объекта пользователь может выбрать сплошной, пунктирный или точечный тип линии. Толщина линии границы меняется путем перемещения движка толщины.

В поле предварительного просмотра объекта отображается внешний вид объекта после произведенных изменений. При открытии ДО свойств для нескольких объектов можно использовать стрелки просмотра этих объектов. Имя объекта отображается в соответствующем текстовом поле.

Независимо от установочных параметров переключатель Active позволяет определять отображение объекта. Активный объект отображается в окне модели со стандартными или определяемые пользователем цветами. Неактивный объект отображается в окне модели белым цветом с серой границей.

Размер выбранных объектов можно изменить, увеличивая или уменьшая процент масштаба. Для восстановления стандартного размера выбранных объектов необходимо выбрать величину 100%. Самое меньшее значение величины масштаба – 10%.

Таким образом, каждый создаваемый объект имеет ряд свойств. Установка этих свойств происходит в ДО свойств объекта. Свойства содержат необходимую информацию об объекте и могут быть заданы по умолчанию или выбраны пользователем.

### 6.3. Организационное моделирование

В организационном описании различают организацию структуры предприятия и организацию процедур выполнения ее БП. В организационном виде моделирования в первую очередь описывают структуру предприятия. Универсальной “совершенной” организационной структуры-прототипа не существует. Оптимальное структурирование организации зависит от различных факторов.

Организационный вид моделирования позволяет, например, всесторонне описать организационно-штатную структуру предприятия. Оптимизация этой структуры является следствием ее анализа и усовершенствования. *Организационно-штатная структура* – это совокупность организационных единиц (структурных подразделений и должностных лиц) и их взаимоотношений в рамках существующих БП. Организационные структуры обычно изображаются в виде организационных диаграмм, где показываются имеющиеся организационные подразделения (как исполнители функций) и их взаимозависимости в соответствии с выбранными

критериями структурирования. Отдельные организационные единицы соединяются связями для указания иерархии. Различают несколько типов организационных единиц и их связей.

На рис.6.3.1 и следующих аналогичных рисунках для каждого типа модели в ARIS 5.0 описаны допустимые типы связей между объектами. В этих рисунках выдержаны следующие ограничения.

1. Объект-источник связи (от него начинается линия связи) изображается всегда слева.

2. Все возможные (для рассматриваемого типа модели) приемники связи (объекты, на которых связь заканчивается) располагаются правее источника связи.

3. Все объекты расположены в произвольном порядке (по их названию), причем источники связи – сверху вниз, а приемники слева направо и при исчерпании места по горизонтали эти объекты располагаются под приемниками связи.

4. Названия или номера связей изображаются над линией связи.

5. Связи с одинаковыми названиями имеют общее начало на объекте-источнике.

6. Если вместо объекта-источника можно поставить другой объект и при этом рисунок не изменится, т. е. объекты приемники и типы связей останутся прежними, то этот факт отмечается на рисунке с соответствующим пояснительным текстом, например, перечислением объектов.

На рис.6.3.1 приведены все возможные объекты-источники и объекты-приемники связи, которые используются в моделях типа *Organizational Chart* (организационная диаграмма) в ARIS 5.0 с Fullfilter (с полным фильтром). Из этого рисунка видно, что модель организационной структуры предприятия может содержать следующие структурные элементы: Организационная единица, Должность или позиция, Личность, Группа и др. Дадим определения этим элементам.

*Организационная единица* (Organizational unit) – это элемент организационной структуры (структурное подразделение), который отвечает за выполнение определенных задач и преследует определенные цели.

*Позиция или должность* (Position) – это наиболее мелкий организационный элемент на предприятии. Обязанности и административные полномочия такого элемента задаются должностными инструкциями.

*Личность* (Person) – это конкретный сотрудник предприятия, имеющий уникальный персональный код.

*Группа* (Group) – это несколько сотрудников, которые вместе работают над конкретной задачей в определенный период времени (например: группа проектирования).

*Внешний сотрудник* (External person) – это лицо,

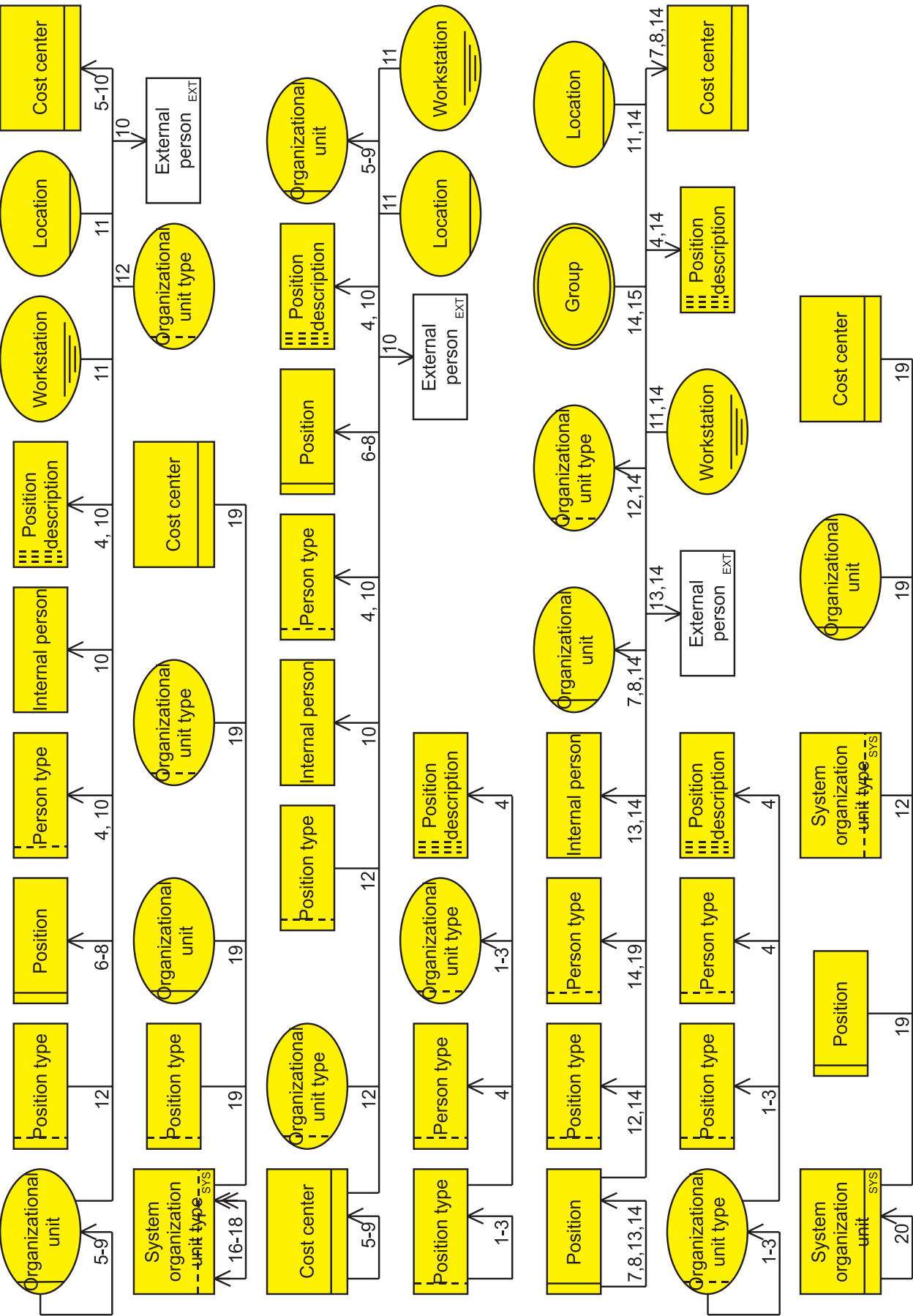


Рис.6.3.1 Объекты и связи, используемые в организационных диаграммах (начало)



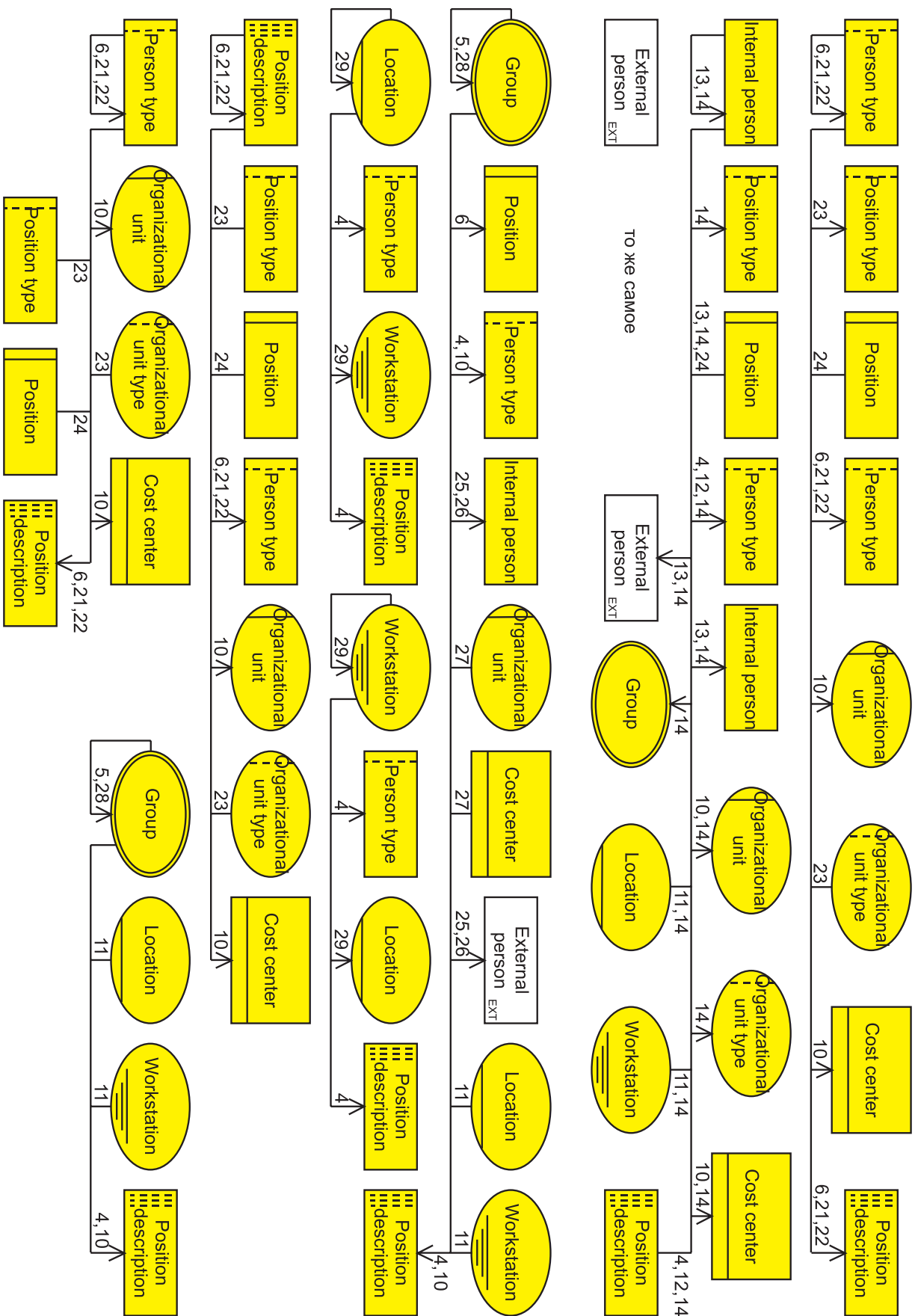


Рис. 6.3.1 Объекты и связи, используемые в организационных диаграммах (окончание)

выполняющее определенные функции в компании, но не являющееся служащим этой компании. Он может назначаться к организационным единицам или функциям, к выполнению которых он привлекается со стороны или за которые он отвечает.

*Сотрудник* (Internal person) является служащим компании и может назначаться к организационным единицам или функциям, которые он выполняет или за которые несет ответственность.

*Тип сотрудника* (Person type) обобщает индивидуальных лиц, имеющих одинаковые характеристики. Эти характеристики могут иметь отношение, например, к одним и тем же полномочиям. Начальники отделов или бригадиры, например, должны следовать определенным правилам и выполнять определенные обязанности, которые достаточно описать только один раз.

*Местонахождение* (Location) определяет физическое положение (местонахождение) организационных единиц, рабочих мест, образцов аппаратных компонентов и технических ресурсов компании. Местонахождение может ссылаться на область, город, предприятие, здание и т. д.

При построении модели организационной структуры между структурными элементами могут устанавливаться, например, следующие типы связей: Ответственный за (is responsible for) или Исполнитель; Технически вышестоящий (is technical superior to) или Технический руководитель; Административно вышестоящий (is disciplinary superior to) или Административный руководитель; Организационный менеджер для (is Organization Manager for) или организатор и др. Все допустимые типы связей в модели типа Organizational Chart представлены в таблице 6.3.1. Номера этих связей указаны на рис.6.3.1.

В качестве примера на рис.6.3.2 приведен фрагмент диаграммы, описывающей организационную структуру государственного управления РФ. На рис.6.3.3 изображена организационная структура филиала ЮУрГУ в г. Миассе. Из нее видно, например, что филиал имеет три факультета, на каждом из которых своя библиотека и компьютерные классы. Количество структурных подразделений на электротехническом факультете (ЭТФ) больше, чем на двух остальных. На рис.6.3.4 детализирована структура деканата ЭТФ, где показаны все субъекты ответственности, работающие в деканате ЭТФ.

Если объекты содержат функциональные обязанности, то организационная схема представляет собой распределение задач фирмы. Для описания отдельных должностей с определенными должностными инструкциями используется объект типа *должность* (Position). За одним подразделением может быть закреп-

лено несколько должностей. За должностями и организационными единицами (ОЕ) можно закрепить лица, фактически занимающие эти должности и являющиеся сотрудниками ОЕ. Объекты Person, Person (m) (мужчина) и Person (f) (женщина) используются для обозначения конкретных лиц. Путем связи конкретного лица с ОЕ указывается, что это лицо выступает в качестве работника этой ОЕ. Путем связи лица с отдельной должностью описывается, что это лицо (сотрудник) в компании занимает эту должность.

ОЕ и лица могут классифицироваться по типам. Это позволяет уточнить, является ли ОЕ отделом (сектором), отделением, группой и т. д. или принадлежат ли лица к типу *руководитель отдела*, *начальник группы* или *начальник проекта*. Для такой классификации можно использовать объекты *Тип организационного подразделения* или *Тип должностного лица*. Типы объектов позволяют описать общие правила БП, в которых принимают участие конкретные ОЕ или должностные лица компании. В процессных цепочках можно уточнить, что только конкретное лицо может выполнять данную функцию или что только определенные типы должностных лиц имеют доступ, например, к информационному объекту.

*Местонахождение* (Location) можно показывать на любом уровне иерархии. Местонахождением могут быть фабрика, здание или индивидуальное рабочее место в комнате. Соответствующее местонахождение может быть отведено для каждой ОЕ уже на этапе определения требований. Поэтому местонахождение ОЕ является важным звеном между этапами определения требований, спецификации и реализации.

Рассмотрим другие типы моделей организационного вида моделирования с полным фильтром в ARIS 5.0.

В процессных моделях можно показать технические ресурсы, которые необходимы для выполнения функций и преобразования материалов. К техническим ресурсам относятся *операционные ресурсы*, *складское оборудование*, *транспортные системы* и *техническое операционное обеспечение*. Эти ресурсы (пока в отрыве от функций) можно описать в организационном виде моделирования. Для этого используется диаграмма типа *Технические ресурсы*, где эти ресурсы можно иерархически упорядочить, присвоить им тип и класс.

*Операционные ресурсы* – это экземпляры различных типов операционных ресурсов, которые доступны для выполнения задач, стоящих перед компанией. Операционные ресурсы часто идентифицируются с помощью различных инвентарных номеров. *Тип операционных ресурсов* представляет совокупность различных операционных ресурсов, которые имеют одинако-

Таблица 6.3.1

| <i>N</i> | <i>Название типа связи</i>   | <i>Перевод названия связи</i>             |
|----------|------------------------------|---|
| 1        | Can be constituent           | Может являться частью                     |
| 2        | Can be technical superior    | Может быть техническим руководителем      |
| 3        | Can be disciplinary superior | Может быть непосредственным руководителем |
| 4        | Performs                     | Формирует                                 |
| 5        | Is superior                  | Имеет в подчинении                        |
| 6        | Is composed of               | Состоит из                                |
| 7        | Is technical superior to     | Является техническим руководителем        |
| 8        | Is disciplinary superior     | Является непосредственным руководителем   |
| 9        | Is responsible for           | Отвечает за                               |
| 10       | Belongs to                   | Принадлежит                               |
| 11       | Is located at                | Располагает                               |
| 12       | Is of type                   | Относится к типу                          |
| 13       | Substitutes for              | Замещает                                  |
| 14       | Is Organization Manager for  | Является организационным управляющим      |
| 15       | Is position of               | Является должностью                       |
| 16       | Is assigned 1:n              | Имеет отношение 1:n                       |
| 17       | Is assigned n:m              | Имеет отношение n:m                       |
| 18       | Is assigned 1:1              | Имеет отношение 1:1                       |
| 19       | Depicts                      | Описывает                                 |
| 20       | Has assigned                 | Имеет назначение                          |
| 21       | Is generalization of         | Является обобщением                       |
| 22       | Is in conflict with          | Находится в противоречии с                |
| 23       | Can belong to                | Может принадлежать                        |
| 24       | Occupies                     | Занимает                                  |
| 25       | Is managed by                | Находится под управлением                 |
| 26       | Has member                   | Имеет в своем составе                     |
| 27       | Is assigned to               | Связан с                                  |
| 28       | Cooperates with              | Взаимодействует с                         |
| 29       | Subsumes                     | Содержит                                  |

вую технологическую базу. Типы операционных ресурсов могут быть объединены в *классы операционных ресурсов*. Их схожесть определяется в соответствии с различными критериями классификации. Следовательно, один тип операционных ресурсов может соответствовать нескольким классам операционных ресурсов. Аналогично складское оборудование, техническое операционное обеспечение и транспортные системы разделяются на экземпляры, типы и классы.

Возможности иерархической организации диаграммы *Технические ресурсы* позволяют описывать структуру технически сложных объектов (заводов, фабрик, конвейеров, роботизированных технологических комплексов и т. д.), а также отображать компоненты сложных производственных объектов и взаимосвязь между ними.

Кроме описания возможностей, в приведенных выше терминах моделирования можно также описать размещение рабочих мест и организационную ответственность за

технические ресурсы. Для этого используются типы объектов *Местоположение*, *Организационная единица*, *Должность* и *Сотрудник*. Эти типы объектов могут быть связаны с типами объектов *Операционные ресурсы*, *Складское оборудование*, *Техническое операционное обеспечение* и *Транспортная система*. Но они должны находиться в самом низу иерархической диаграммы. На рис.6.3.5 показан общий вид такой диаграммы. Здесь на месте Операционных ресурсов, их типов и классов может стоять любой из соответствующих объектов для складского оборудования, транспортных систем и т. д. На рис.6.3.6 приведены все возможные объекты-источники и объекты-приемники связей, которые используются в моделях типа *Технические ресурсы*. Возможные типы связей приведены в таблице 6.3.2.

Еще один тип модели, который используется на этапе анализа требований в организационном моделировании, называется *Shift calendar* (*Календарь смен*).

*Календарь смен* – это многоуровневая объектная

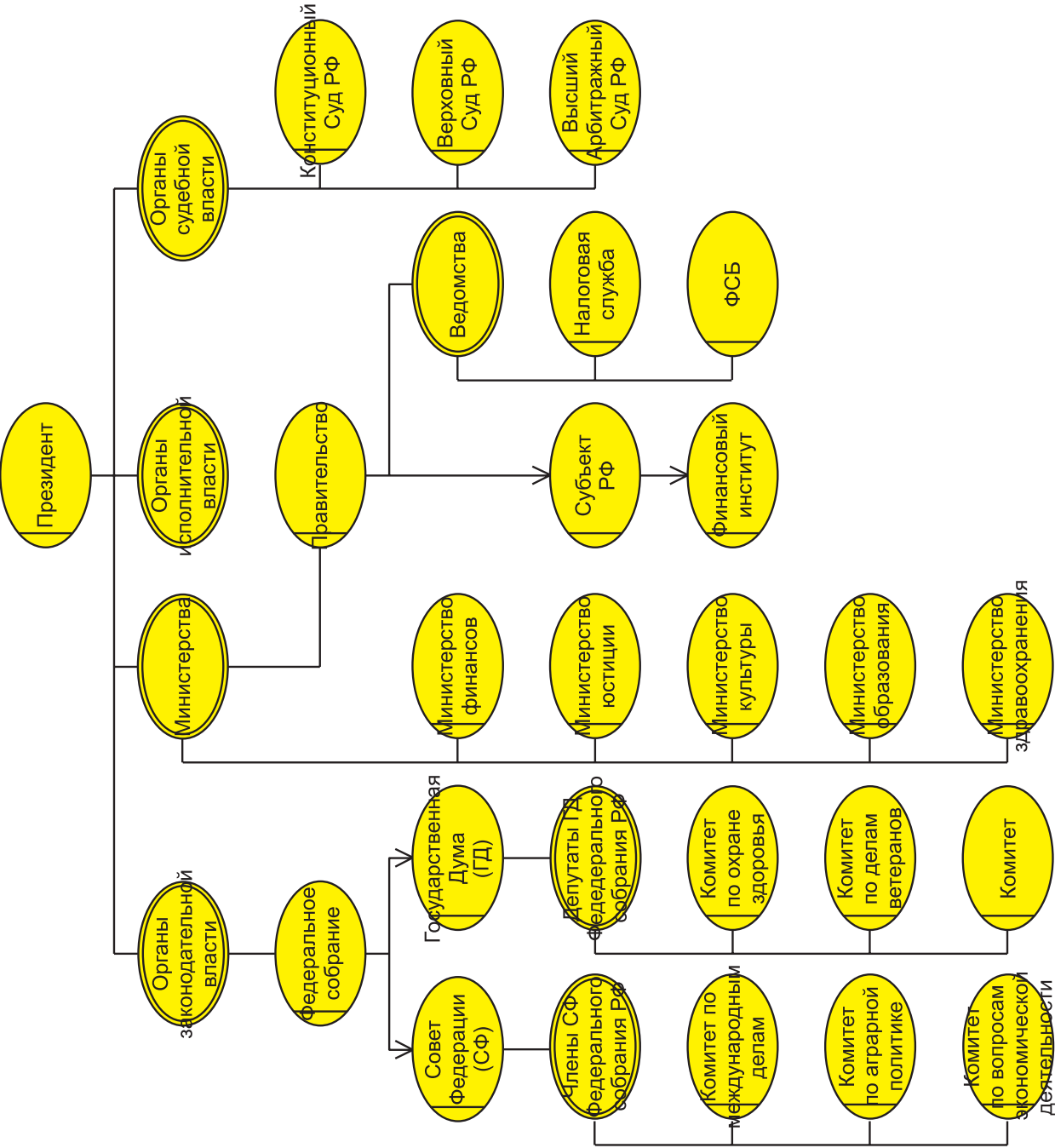


Рис.6.3.2 Фрагмент организационной структуры государственного управления РФ



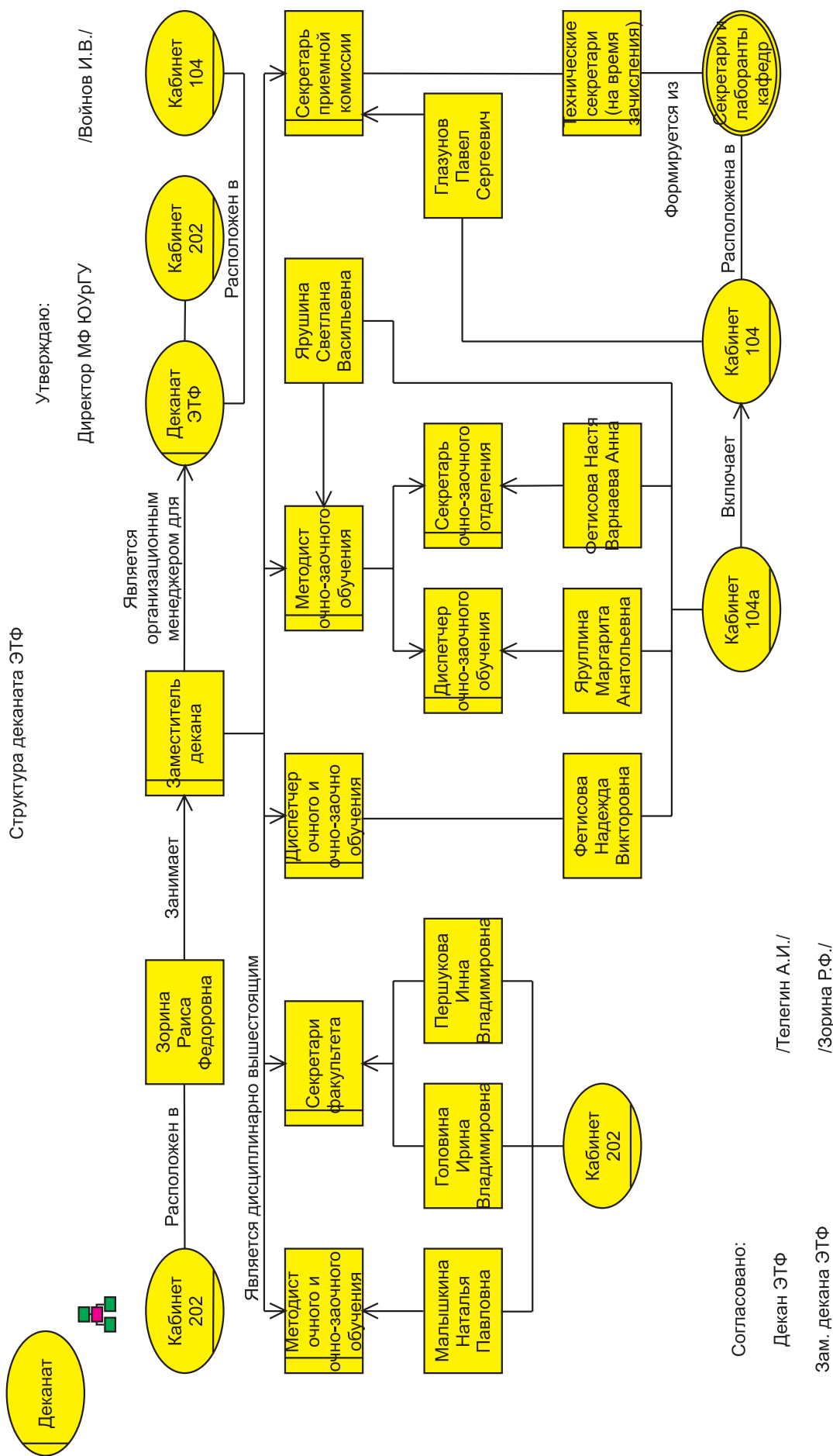


Рис.6.3.4 Организационная диаграмма деканата электротехнического факультета ЮУрГУ (в г. Миассе)



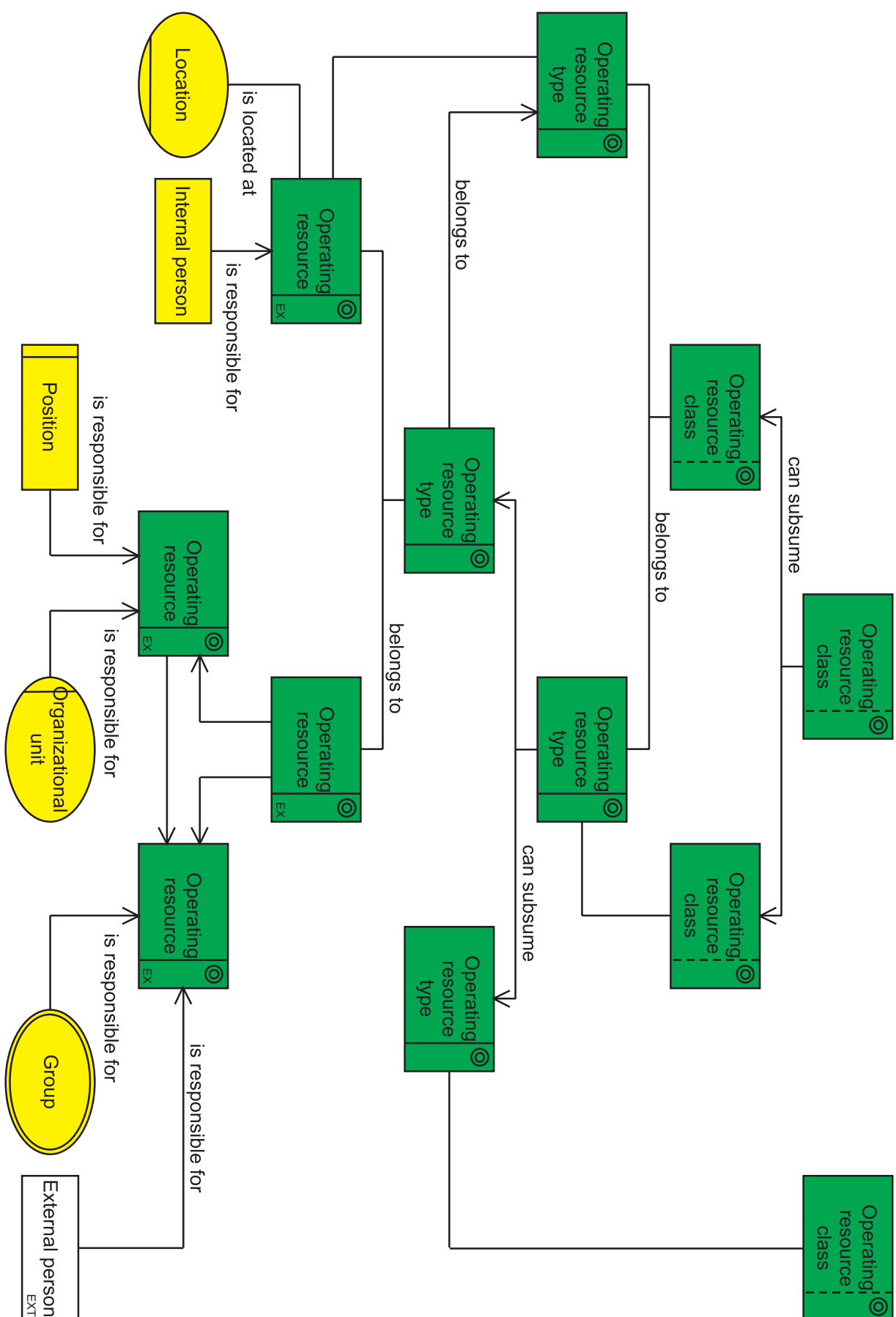


Рис.6.6.3.5 Общий вид диаграммы типа “Технические ресурсы”

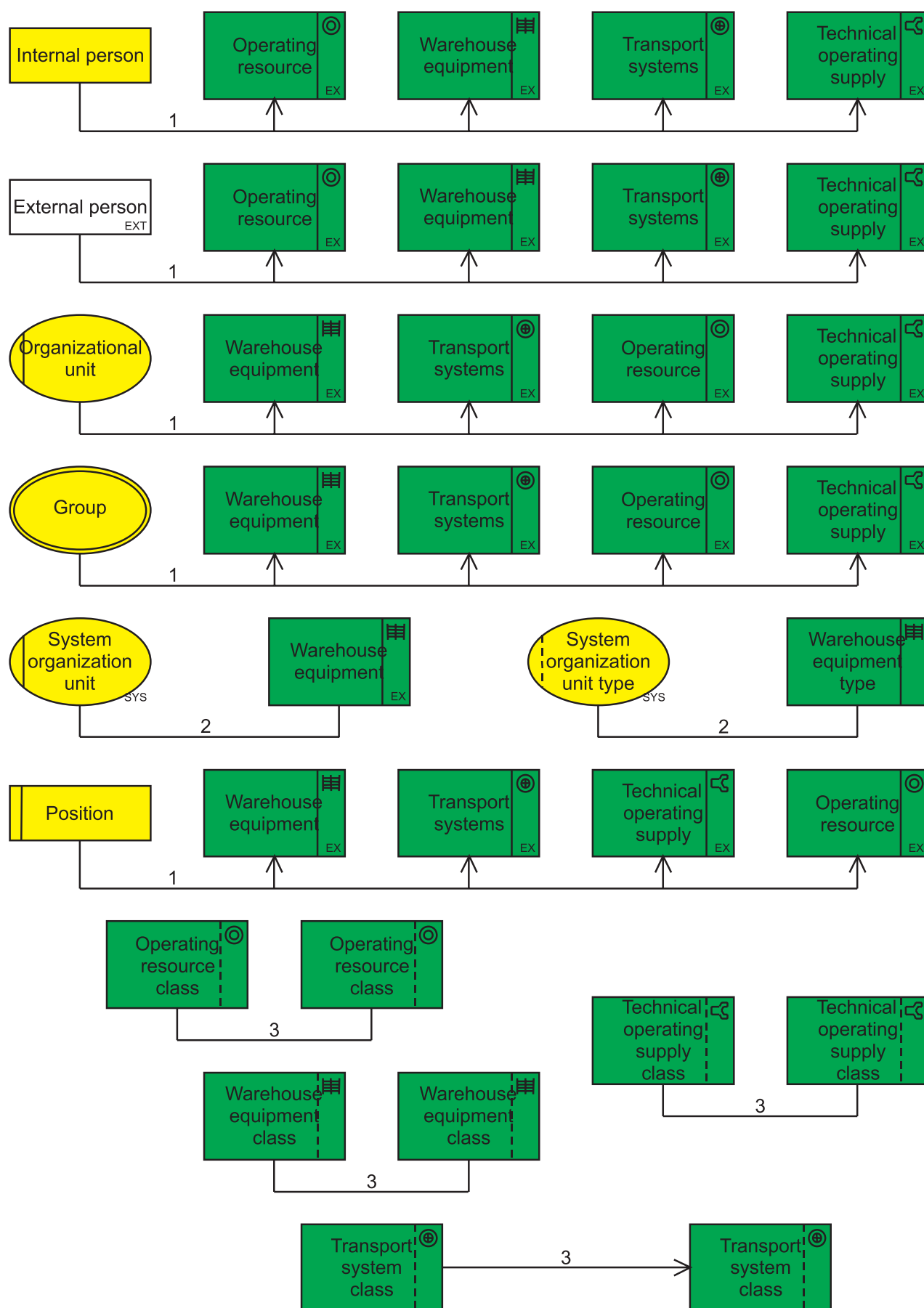


Рис.6.3.6 Объекты и связи в моделях типа Technical resources (начало)

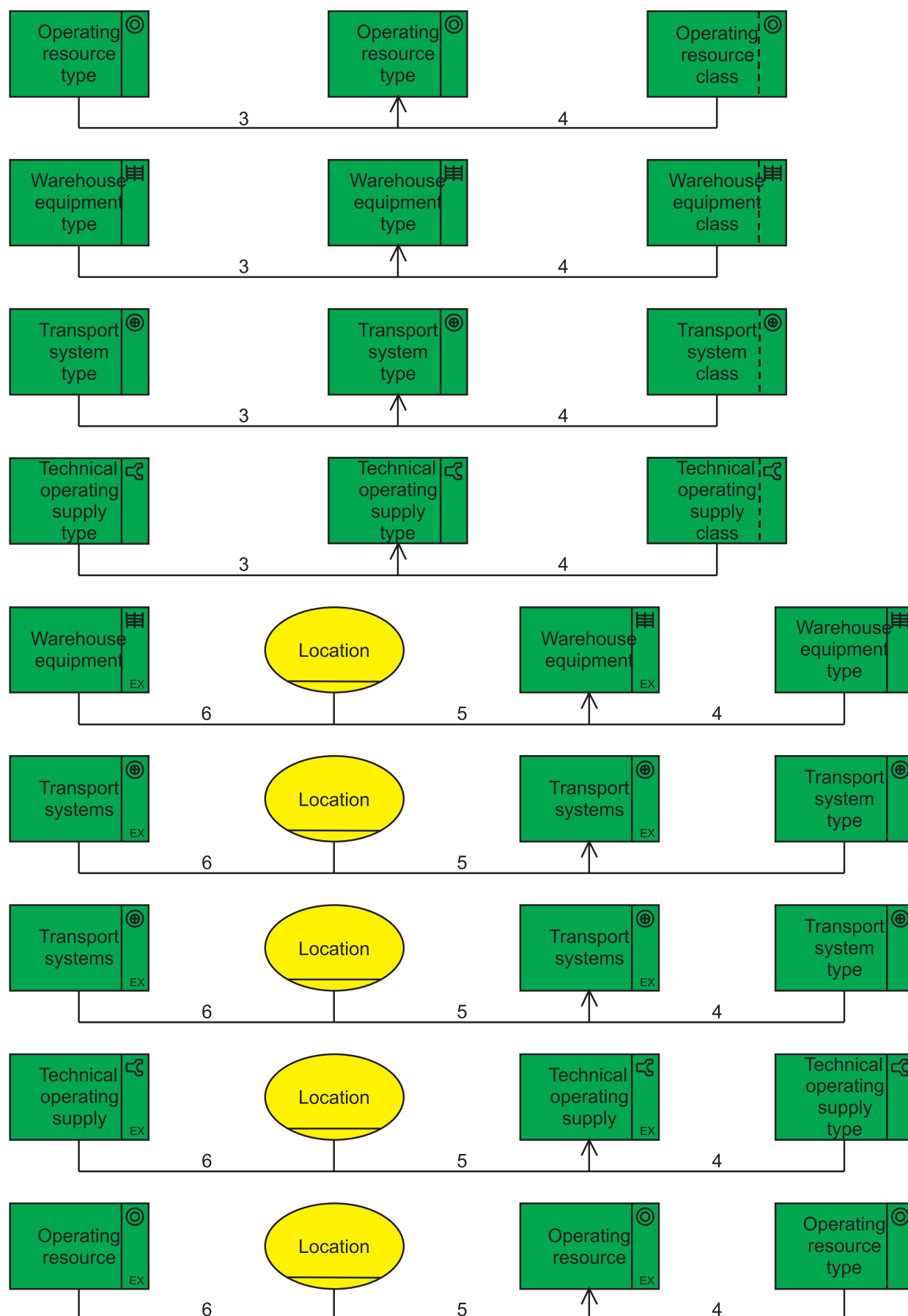


Рис.6.3.6 Объекты и связи в моделях типа Technical resources (окончание)

Таблица 6.3.2

| N | Название типа связи | Перевод названия связи                   |
|---|---------------------|--|
| 1 | Is responsible for  | Является ответственным за                |
| 2 | Depicts             | Описывает                                |
| 3 | Can subsumes        | Может являться обобщающей категорией для |
| 4 | Belongs to          | Принадлежит                              |
| 5 | Subsumes            | Содержит, объединяет                     |
| 6 | Is located at       | Располагается                            |

модель. Он назначается субъектам в организационной схеме или диаграмме eEPC и в общем случае может быть связан с любым человеческим или материальным ресурсом. Если существует иерархия человеческих ресурсов, календарь смен всегда располагается на самом нижнем уровне иерархии.

В самом календаре смен также на самом нижнем уровне иерархии находятся объекты типа *Перерыв*. Это ежедневный интервал времени в рамках смены, в течение которого работа не выполняется. Перерыв описывается относительным временем его начала и продолжительностью, причем относительное время начала всегда ссылается на смену, с которой данный перерыв связан.

Следующий уровень в иерархии включает объекты типа *Смена*. Это ежедневный интервал времени, в течение которого выполняется работа. Смена описывается относительным временем начала и продолжительностью. Смена может иметь несколько перерывов. Например, на электротехническом факультете ЮУрГУ обучение ведется в три смены (утренняя, дневная и вечерняя), которые повторяются каждые 24 часа.

*Цикл смен* – это еженедельный интервал времени или интервал, длящийся несколько дней, в течение которого выполняется работа. Цикл смен определяет дни, когда некоторая смена работает, а когда нет. Сам цикл смен определяется относительным временем начала цикла и его продолжительностью. Если цикл смен повторяется непрерывно, то этот факт описывается атрибутом *Период*, который определяет, как часто данный цикл должен повторяться. Циклы смен часто охватывают период от одной до двух недель. Сотрудник может работать в утреннюю смену в течение одной недели и в дневную в течение другой. Эта последовательность может повторяться непрерывно через циклы смен.

*План смен* – это совокупность всех циклов и их смен, которая описывает рабочие часы некоторого ресурса. В *календаре смен* описывается только то, что повторяется периодически. Поэтому болезни, праздники и другие дни, в течение которых работа не выполняется, в это описание не включаются.

На этапе спецификации в организационном виде

используется диаграмма типа *Топология сети*. Она дополняет организационную диаграмму компании описанием коммуникационной и информационной структуры этой компании. Путем моделирования организационной структуры компании создается базис для описания сетевой топологии. Для определения сетевой топологии необходимы соединения (линии) сети и узлы сети, которые располагаются в определенных местах компании.

Диаграмма топологии сети может содержать различные *типы сетей*, эксплуатируемых в компании. Тип сети представляет отдельные *экземпляры сети*, которые базируются на одной и той же технологии. Каждый тип сети может быть связан с возможными *типами узлов сети* и *типами соединений сети*. Каждый тип соединения сети может включать информацию о типах узлов сети, в которых она оканчивается. *Типы компонент аппаратного обеспечения (АО)* относятся в этом случае или к оборудованию сети, реализующему сетевые структуры, или к типам компонент оборудования, которые могут быть подсоединены к типам узлов сети. Типы компонент АО – это отдельные компоненты оборудования, которые базируются на одной и той же технологии. Типы компонент АО могут располагаться в любом месте иерархии.

Примером типа соединения может быть отдельный тип сетевого кабеля. Кроме того, можно показать, какие типы компонент АО к какому типу узла сети могут быть подсоединены. Типы компонент АО могут иерархически связываться с другими типами компонент, которые необходимы для создания типов узлов.

Связь между топологией сети и объектами, определяемыми на этапе формулировки требований, устанавливается либо через ОЕ или должности, которые ответственны за каждый тип компонент АО, либо через место расположения данного типа сети, типа узла сети, тип соединения и тип компонент аппаратуры. Таким образом, можно связать сетевые узлы с рабочими местами ОЕ. Поэтому местоположение – это центральное звено между сформулированными для организационной модели требованиями и спецификацией проекта.

На этапе описания реализации используется тип мо-

дели *Диаграмма сети*, которая описывает реализацию топологии сети. Здесь показывают конкретные сети компании, их узлы и соединения. Можно описать точное местоположение каждой сети, ее узла и соединения. Можно зарегистрировать (инвентарным номером) компоненту АО, представленную в каждом соединении и узле сети.

На рис.6.3.7, 6.3.8 изображены все возможные объекты-источники и объекты-приемники, а также номера типов тех связей, которые используются в моделях типа *Топология сети* и *Диаграмма сети*. Названия связей для этих моделей представлены соответственно в таблицах 6.3.3, 6.3.4.

Таблица 6.3.3

| <i>N</i> | <i>Название типа связи</i> | <i>Перевод названия связи</i>            |
|----------|----------------------------|--|
| 1        | Can subsume                | может являться обобщающей категорией для |
| 2        | Can be connected to        | может быть подсоединен к                 |
| 3        | Is used in                 | используется в                           |
| 4        | Can occur                  | может относиться к                       |
| 5        | Can be realized by         | может быть реализован с использованием   |
| 6        | Can consists (hor.) of     | может состоять из (горизонт.)            |
| 7        | Can consists (vert.) of    | может состоять из (верт.)                |
| 8        | Can end in                 | может заканчиваться в                    |
| 9        | Belongs to class           | принадлежит классу                       |
| 10       | Is predecessor of          | предшествует                             |
| 11       | Subsumes                   | содержит (объединяет)                    |
| 12       | Can realize                | может являться реализацией               |
| 13       | Can be responsible for     | может являться ответственным за          |

Таблица 6.3.4

| <i>N</i> | <i>Название типа связи</i>  | <i>Перевод названия связи</i>              |
|----------|-----------------------------|--|
| 1        | Subsumes                    | содержит (объединяет)                      |
| 2        | Is of type                  | относится к типу                           |
| 3        | Belongs to                  | принадлежит                                |
| 4        | Is implemented by           | реализуется в виде (или реализован кем то) |
| 5        | Is located at               | располагается                              |
| 6        | Consists (hor.) of section  | состоит из секций (горизон.)               |
| 7        | Consists (vert.) of section | состоит из секций (верт.)                  |
| 8        | Ends in                     | оканчивается в                             |

#### 6.4. Моделирование данных

Модели, описывающие структуру данных, явились исторически первым средством отображения формулировок требований. В настоящее время существует несколько способов формального описания структур данных. Из них модель ERM, которую изобрел Петер Пин-Шен Чен, является доминирующей. На этой модели основано большинство программных средств проектирования структур данных. В CASE-ориентированной литературе эту модель называют по-разному, например, ER-модель (entity-relationship model), модель сущность-отношение (Entity – сущность, Relationship – отношение), модель сущность-связь.

ER-модель изобретена с целью создания единой методики семантического описания данных. Чен ввел специ-

альный диаграммный метод как средство проектирования БД. Он привел примеры описания и проектирования БД с использованием этой модели и диаграммного метода. В настоящее время ER-модель используется в качестве основы для унификации различных представлений данных. Хорошо разработана методика порождения существующих представлений данных из ER-модели, обладающей большинством преимуществ этих представлений. Поэтому ER-модель можно рассматривать как обобщение и расширение существующих моделей.

При описании структур данных, а также входных и выходных информационных потоков БП в ARIS используются различные типы моделей данных, но чаще всего используется модель Чена, т. е. модель типа ERM или коротко *ER-модель*. Эта модель содержит такие объекты, как *сущность*, *отношение*, *атрибуты* и т. д., которые описы-

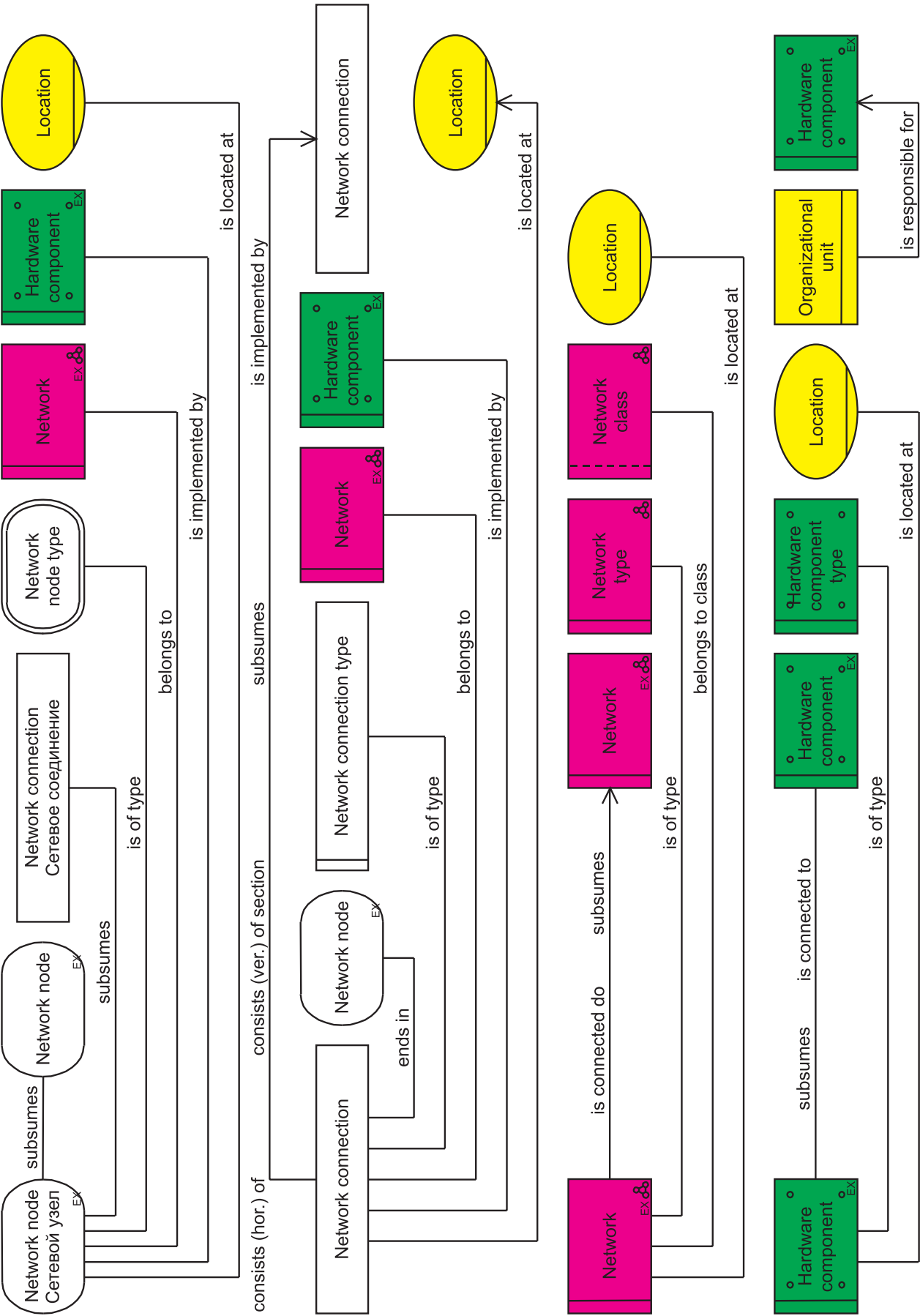


Рис.6.3.7 Допустимые объекты и связи в диаграммах типа “Диаграмма сети”





вают начальные и конечные события в цепочке БП и в целом состояние инфраструктуры, связанной с БП.

Как видно из названия, модель сущность-отношение представляет данные в виде набора сущностей, связанных друг с другом отношениями. Под сущностью понимается реальный или абстрактный объект, представляющий интерес с точки зрения рассматриваемого круга задач предприятия. Отношение – это логическая связь между двумя и более сущностями. Качественные и количественные характеристики сущностей и отношений сосредоточены в их атрибутах. Например, сущность *клиент* может быть более точно определена атрибутами *имя*, *фамилия* и *адрес*.

На ER-диаграммах сущность изображается прямоугольником, содержащим имя сущности, а отношение ромбом, связанным линией с каждой из взаимодействующих сущностей. Числа над линиями означают степень отношения. ER-диаграмма должна подчиняться следующим правилам: каждая сущность, отношение и их атрибуты должны иметь имя; имена сущностей и отношений должны быть уникальными в рамках модели; имена атрибутов должны быть уникальными в рамках сущности; каждый атрибут должен иметь определение типа данных; сущность должна иметь ключевой атрибут и т. д.

Построение ER-модели начинается с изображения сущностей и тех отношений между ними, которые возникают в нашем воображении. Затем вводится структура этой информации, в которой сущности, отношения и атрибуты представляются данными.

Можно сказать, что сущность – это предмет, который может быть идентифицирован некоторым способом, отличающим его от других предметов. Конкретные человек, компания или событие являются примерами сущности. Под отношением можно понимать ассоциацию или логическую связь между сущностями. Следовательно, существование отношений зависит от существования сущностей.

БД предприятия содержит информацию о тех сущностях и их отношениях, которые представляют интерес для этого предприятия. Причем, нет необходимости хранить исчерпывающую информацию, но важно хранить всю потенциально доступную и важную информацию о сущностях и их отношениях. В ER-моделях рассматривают только те сущности, отношения и информацию о них, которые представляют интерес и должны войти в БД предприятия.

В ARIS модель сущность-отношение Чена расширена с помощью операторов проектирования, которые обеспечивают формальную поддержку процесса создания модели данных и ее связи с другими моделями

в рамках процессного вида моделирования. Например, анализ условий выполнения БП с точки зрения их структур данных помогает описывать условия и события, при которых выполняются функции БП. Выделяют четыре основных оператора проектирования: *классификацию*, *обобщение*, *агрегацию* и *группировку*.

При помощи *классификации* близкие друг к другу (идентичные) объекты (сущности, отношения и т. д.) объединяются в соответствии с некоторым признаком в один тип (тип сущности, тип отношения и т. д.). Каждая сущность относится к некоторому, отличному от других набору сущностей, т. е. к некоторому типу сущностей. Например, клиент *Иванов* и клиент *Петров* вместе образуют сущность типа *клиент*, а изделие 2314 вместе с изделием 2315 образуют сущность типа *изделие*. С каждым набором сущностей, т. е. с типом, связывается предикат (идея, мотивировка, принцип отбора), который позволяет проверить, принадлежит ли сущность данному набору. Например, *отец* и *сын* – это две сущности из набора сущностей *мужчины*. Если мы знаем, что сущность относится к набору сущностей *мужчины*, то мы также знаем, что эта сущность обладает свойствами (атрибутами), общими с любой сущностью типа *мужчины*. Экземпляром типа сущности является сущность. Сущности одного и того же типа описываются одними и теми же атрибутами. Заметим, что разные наборы сущностей могут пересекаться. Например, сущность, принадлежащая типу сущностей *мужчины*, может принадлежать также и к типу сущностей *люди*.

Аналогично идентичные отношения, сгруппированные в набор, определяют *тип отношения*. Например, тип отношения между сущностью *поставщик* и сущностью *деталь* может быть *снабжает*. В ER-модели типы отношений изображаются ромбами, которые связываются с типами сущностей, изображаемых прямоугольниками.

Информацию о сущностях и отношениях получают путем наблюдения или измерения и выражают множеством пар *атрибут-значение*. Значения классифицируются в различные *наборы значений*, такие, как *возраст*, *цвет*, *имя*, *фамилия*. С каждым набором значений связывается предикат для проверки того, принадлежит ли значение этому набору. *Атрибуты* описывают типы сущностей и отношений. *Экземпляры атрибутов* – это фактические значения атрибутов, присвоенные отдельным сущностям и отношениям. Например, *клиент* может быть точнее описан посредством экземпляров атрибутов *Петров*, *Иван*, *Москва*, если у типа сущности *клиент* есть атрибуты *фамилия*, *имя* и *город проживания*.

Атрибуты изображаются эллипсами. На рис.6.4.1 приведены атрибуты для типа сущности *клиент*. Здесь

эллипсом с двойной границей обозначен ключевой (уникальный) атрибут.

Часто достаточно сложно определить различие между типами сущностей и атрибутами. Например, *адрес клиента* может пониматься как сущность, а не как атрибут сущности *клиент*. В этом случае можно было бы ввести новый тип сущности – *адрес*, который имел бы собственное отношение к сущности *клиент*, например, под названием *проживание*. Для преодоления этой неоднозначности следует помнить, что атрибуты не могут иметь атрибуты. Поэтому, если в ER-модели создается объект и предполагается, что он будет в дальнейшем описан атрибутами, то этот объект должен быть типом сущности. Другой важный критерий при определении сущности – будет ли рассматриваемый объект взаимосвязан с другими типами сущностей или нет. Если такая связь предполагается, то этот объект также является типом сущности, а не атрибутом.

Типы отношений часто могут читаться только в одном направлении, при котором связи между сущностями и их отношениями имеют смысл. Например, отношение *поставщик снабжает деталями* имеет смысл. Читая это отношение справа налево, получим: *детали снабжают поставщика*, что не имеет смысла.

Типы отношений классифицируются по степени сложности и по числу типов сущностей. По числу типов связанных сущностей различают *унарные*, *бинарные* и *n-арные* отношения. На рис.6.4.2 представлены бинарные отношения, а на рис.6.4.3 изображено унарное отношение.

Степень сложности (*мощность отношения*) указывает, сколько сущностей одного типа связаны с одной сущностью другого типа. Различаются четыре типа отношений: *отношение 1:1*, *отношение 1:n*, *отношение n:1*, *отношение n:m*. При отношении 1:1 только одна сущность из второго типа связывается (может вступать в отношения) с каждой сущностью из первого типа. При отношении 1:n только одна сущность из второго типа связана с каждой сущностью из первого, но n сущностей из первого типа связаны с каждой сущностью из второго. Отношение n:1 отображает ту же ситуацию, но в обратном порядке. При отношении n:m несколько сущностей из второго типа связаны с одной сущностью из первого типа и наоборот. На рис.6.4.2 приведены соответствующие примеры.

Отношения могут существовать между сущностями (экземплярами) одного типа. Такие отношения называются *рекурсивными*. В этом случае тип сущности и тип отношения связываются друг с другом двумя линиями (соединениями). Для того, чтобы дифференцировать эти соединения, можно указать над связью ее

старшинство или подчиненность. Пример рекурсивного отношения приведен на рис.6.4.3. Здесь показано, что главная деталь состоит из различных подчиненных деталей. С другой стороны, подчиненная деталь может быть компонентой в различных главных деталях.

Диапазон значений атрибутов, описывающих сущности и их отношения, называется *областью*. Соотнесение элементов области элементам сущности или типам отношений также описывается отношениями. Они могут быть представлены линиями, которые помечаются именами атрибутов. Отношение типа 1:1 должно существовать между одним типом сущности и, по крайней мере, одной областью. Значения этой области могут однозначно идентифицировать отдельные сущности. Они называются *ключевыми атрибутами* данного типа сущности. В примере на рис.6.4.4 отдельные сущности типа *клиент* однозначно идентифицированы ключевым атрибутом *номер клиента*. Отношения идентифицируются слиянием ключевых атрибутов связанных сущностей. Таким образом, ключевые атрибуты отношения типа *находится по* – это *номер клиента* и *номер адреса*.

Перейдем к рассмотрению операторов проектирования *Обобщение/Специализация*. При обобщении аналогичные типы сущностей группируются под одним, более старшим, типом сущности. Например, на рис.6.4.5 тип сущности *клиент* и тип сущности *поставщик* объединены в тип сущности *участник БП*. Свойства (атрибуты), которые являются общими для исходных сущностей, присваиваются обобщенной сущности. Образование нового типа сущности *участник БП* представляется графически в виде треугольника, который называется *отношением обобщение/специализация* и может читаться словом – *есть*. Поэтому можно сказать *клиент* и *поставщик* есть *участник БП*.

Под специализацией понимается разделение некоторого общего множества объектов (например, сущностей) на подмножества. Оператор специализации является инверсным (обратным) по отношению к оператору обобщения. Например, при специализации сущность *участник БП* разделяется на сущность *клиент* и сущность *поставщик*. Специализированные сущности наследуют свойства (атрибуты) обобщенных сущностей. Кроме наследованных, специализированные типы сущностей могут обладать также собственными атрибутами. Графически специализация и обобщение представляются одинаково (треугольником). Поэтому в линиях связи на рисунке не изображаются стрелки, указывающие направление связи.

Специализация поддерживает подход описания структуры данных “сверху вниз”, обобщение же исполь-



Рис.6.4.1 Атрибуты типа сущности "Клиент"

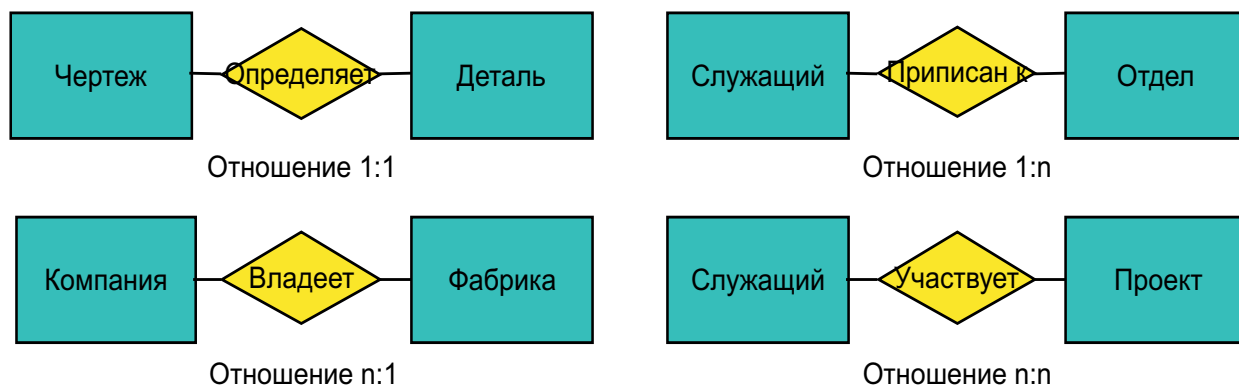


Рис.6.4.2 Примеры мощности типов отношений в ER-модели

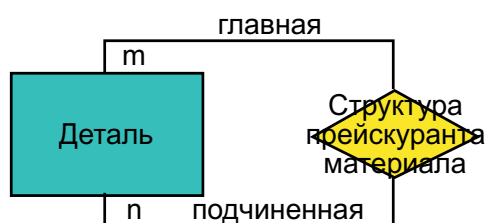


Рис.6.4.3 Рекурсивное отношение

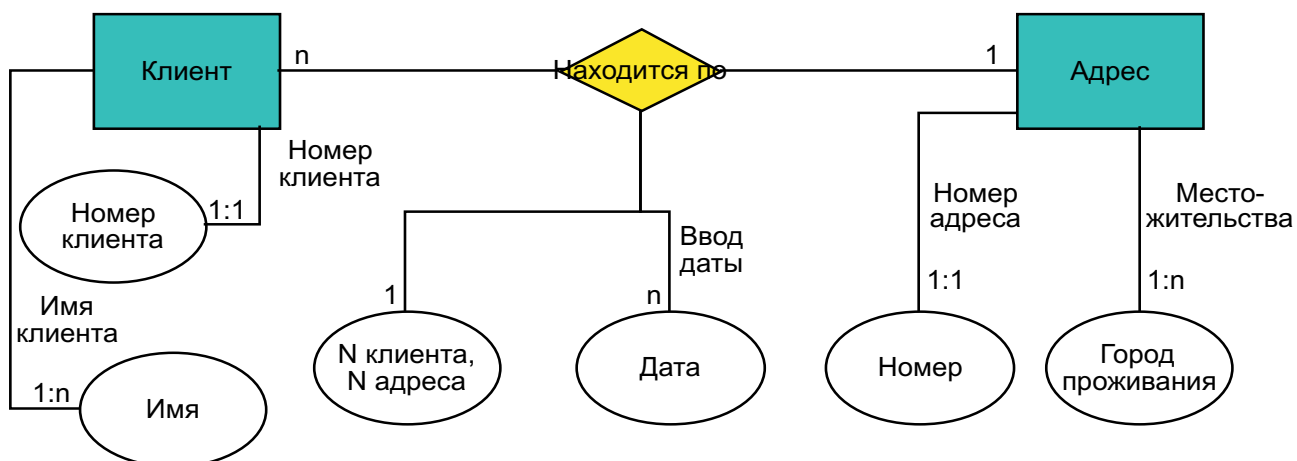


Рис.6.4.4 Определение атрибутов в ER-модели

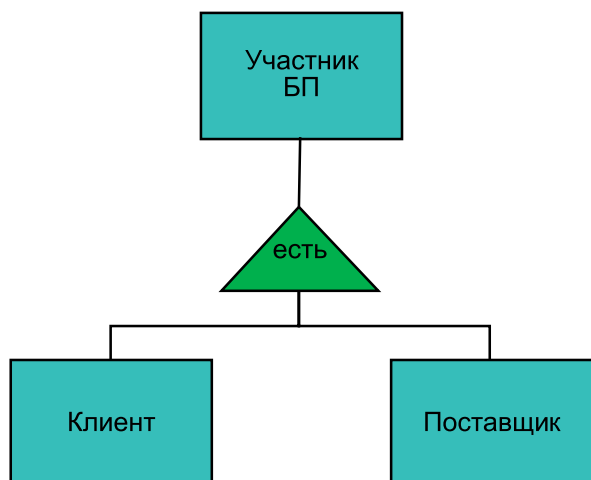


Рис. 6.4.5 Оператор обобщения/специализации

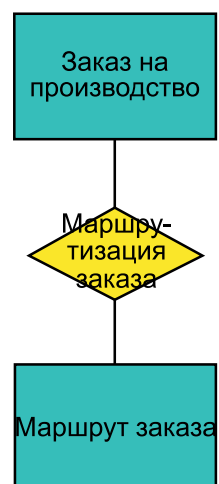


Рис. 6.4.6 Оператор агрегации

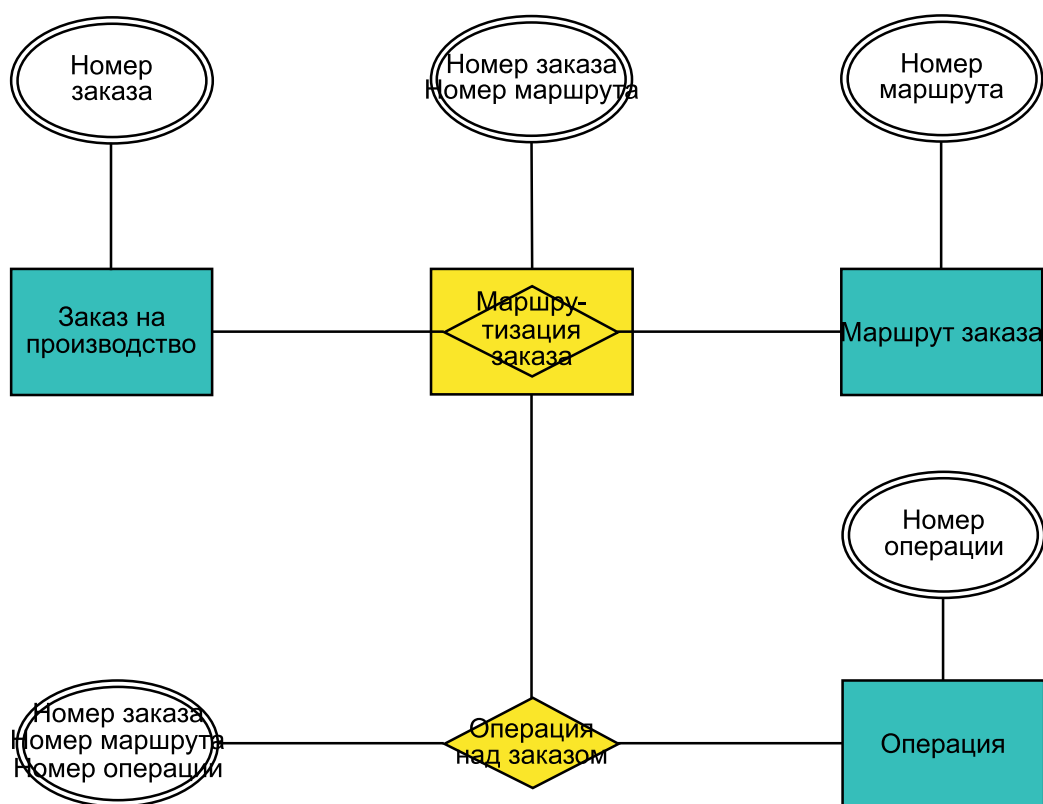


Рис. 6.4.7 Агрегация с реинтерпретацией типа отношений

зуется при подходе “снизу вверх”.

Используют четыре возможных случая оператора обобщение/специализация: *раздельная полная, раздельная неполная, нераздельная полная и нераздельная неполная*. Для того, чтобы их различать, отметим, что подмножества неразделимы, если экземпляр одного из них может быть частью обоих подмножеств. Например, некоторый *клиент* в то же самое время может быть и *поставщиком*. Если экземпляр ассоциируется только с одним подмножеством, то подмножества разделимы. И еще, например, полная специализация имеет место, когда все специализированные типы сущностей, возможные для одного критерия специализации, входят в состав одного обобщенного типа сущности. Например, тип сущности *человек* может быть разбит на типы сущностей *мужчина* и *женщина*. Это полная специализация, т. к. в качестве критерия специализации выбран пол.

Еще один оператор проектирования *агрегация* описывает формирование нового типа объекта с помощью комбинации существующих типов объектов, где новый тип объекта может иметь новые свойства. В ER-модели агрегация представляется как формирование типов отношений (рис.6.4.6) из сущностей. Агрегация типа сущности *заказ на производство* и типа сущности *маршрут заказа* создает новый объект, а именно отношение *маршрутизация заказа*.

Оператор агрегации также применяется к отношениям. В этом случае существующий тип отношения трактуется как тип сущности и, таким образом, может стать отправной точкой для создания другого нового отношения. На рис.6.4.7 первая агрегация используется для формирования типа отношения *Маршрутизация заказа* из типов сущностей *Заказ на производство* и *Маршрут заказа*. Ключевые атрибуты *номер заказа* на производство и *номер маршрута* образуют сложный ключевой атрибут для отношения *Маршрутизация заказа*. Теперь многочисленные операции могут быть связаны с *Маршрутизацией заказа*. Следовательно, отношение *Операция над заказом* формируется между типом отношения *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.6.4.7 это иллюстрирует ромб, заключенный в прямоугольник. В конце концов, этот реинтерпретированный 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, означаю-

щие новые отношения по реинтерпретированным типам отношений, подходят к сторонам прямоугольника, окружающим ромб, а не к самому ромбу.

В процессе применения оператора *группировка* формируются группы, составленные из элементов некоторого множества сущностей. Например, все составные части *Оборудования* объединяются в *Группу оборудования*, т. е. независимый объект, который может быть описан более точно с помощью дополнительных атрибутов (наименование группы оборудования, число деталей оборудования), нехарактерных для отдельных частей оборудования. Другим примером может служить группировка рабочих мест в отделы или объединение элементов, связанных с обработкой заказа, в группу *Заказ*.

В сложных моделях рекомендуется выделять *классы данных*, которые объединяют логически связанные типы сущностей и отношений в одно целое. Это позволяет упростить описание сложных моделей данных. *Класс данных* – это логическая модель некоторых типов сущностей и отношений. Например, кластер (сложный объект) *Заказ клиента* содержит типы сущностей *Клиент*, *Время*, *Изделие* и отношения *Предмет заказа клиента*, *Позиция заказа клиента*. Кластеры данных состоят не только из типов сущностей и отношений, но и из других кластеров данных. В отличие от типов сущностей и отношений кластеры данных могут быть легко встроены в иерархию, что позволяет в процессе создания модели данных поддерживать технологию моделирования “сверху вниз”. Использование кластеров данных может быть полезно и при объединении подмоделей в более сложную модель по технологии “снизу-вверх”.

Перейдем к краткому рассмотрению других типов моделей данных в ARIS 5.0 с полным фильтром.

Модели данных типа *eERM (расширенные модели “сущность-отношение”)* очень часто имеют довольно сложную структуру. Диаграммы, включающие атрибуты сущностей и отношений, очень громоздки и теряют свою наглядность. С помощью моделей типа *eERM attribute allocation diagram (Диаграмма распределения eERM атрибутов)* можно отдельно описать атрибуты для каждого типа сущности и отношения. В эту диаграмму можно включить тип объекта из *диаграммы eERM* (тип сущности или тип отношения). Таким образом моделируется распределение атрибутов по объектам eERM, где можно увидеть, является ли атрибут *ключевым (K attribute ERM)*, *описательным (D attribute ERM)* или *внешним ключом (EK attribute ERM)*. Кроме представления и распределения отдельных атрибутов eERM, в этом типе диаграммы можно также отобразить группу типов атрибутов и их распределение. *Группа типов атрибутов (Attribute type group)* представляет объедине-



ние атрибутов одного типа сущности из диаграммы eERM, которые семантически близко связаны. Это позволяет создать группу атрибутов, содержащую все атрибуты eERM, которые вместе образуют, например, вторичный ключ.

Кроме рассмотренного способа моделирования данных (модели типа eERM), существует несколько альтернативных форм представления ER-моделей. Например, модель *SERM* (структурная модель “сущность-отношение”), разработанная Синцом, а также модель данных *IEF*, которая соответствует обозначениям, принятым при моделировании данных с помощью инструментария CASE Information Engineering Facility (IEF) фирмы Texas Instruments Inc. К альтернативной форме описания модели данных относится также нотация семантической модели данных *SeDaM* (*Semantic Data Model*).

В моделировании больших систем часто встречаются затруднения, связанные с многочисленностью терминов и их различной трактовкой. Например, то, что понимается под термином *заказ* в отделе закупок, может отличаться от того, что под этим подразумевают сотрудники производственного отдела. Для устранения этой неоднозначности и в целом для наведения порядка в используемых понятиях и их систематизации служит модель типа *Technical Term Models* (*Модели технических терминов*) (смотри рис. 6.4.8). При введении соответствующей терминологической модели (диаграммы) в компании и ее отделах используемые термины становятся более понятными и однозначно воспринимаемыми. Они могут использоваться и в других диаграммах, например, в eEPC для представления входных и выходных данных функции. Технические термины могут быть взаимосвязаны и иерархически упорядочены. 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. 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. Тип связи *can be* (*может являться*) означает, что один из терминов может являться экземпляром из множества значений другого термина. Тип связи *is specimen of* (*является экземпляром*) предназначен для отражения возможных экземпляров терминов.

С помощью модели типа *Material diagram* (*Диаграммы материалов*) можно описывать типы используемых на предприятии материалов, упорядочивать их иерархически и классифицировать по классам материалов. *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*. При этом один тип материала может относиться к нескольким классам материалов. Типы материалов могут быть связаны с *типами упаковочных материалов*. Это означает, что определенные типы материалов могут перемещаться, только если используется конкретный тип упаковочных материалов. *Упаковочный материал* также может быть определен, классифицирован и иерархически упорядочен. Это позволяет, например, описать структуру и ограничения сложных упаковочных товарных комплексов. *Тип упаковочного материала* — это совокупность отдельных упаковочных материалов, имеющих одинаковые характеристики. *Типы упаковочных материалов* могут быть объединены в *классы упаковочных материалов*. На рис. 6.4.9 представлены четыре фрагмента, в которых слева стоит источник связи, справа — приемник связи, а над линией связи записано ее название.

Для определения стоимости объекта или процесса по методу пооперационного исчисления стоимости (ABC) применяется модель типа *CD Diagram* (*Диаграмма стоимостных драйверов*). Она описывает иерархию стоимостных драйверов, т. е. иерархическую взаимосвязь измеряемых (вычисляемых) величин, предназначенных для оценки стоимости отдельных операций (функций) в рамках БП. В этой диаграмме используется единственный тип объекта *Cost driver*, между экземплярами которого возможна связь типа *determines volume of* (*определяет объем*). С этой моделью тесно связана модель типа *Cost category diagram* (*Диаграмма стоимостных категорий*). Она также используется для вычисления стоимости по методу ABC. Но сначала строится иерархия стоимостных категорий. *Стоимостные категории* служат для систематического структурирования всей стоимостной информации, которая появляется в результате создания или оценки стоимостных драйверов (результатов). В этой диаграмме используется единственный тип объекта *Cost category*, между экземплярами которого возможна связь типа *is superior* (*имеет в*

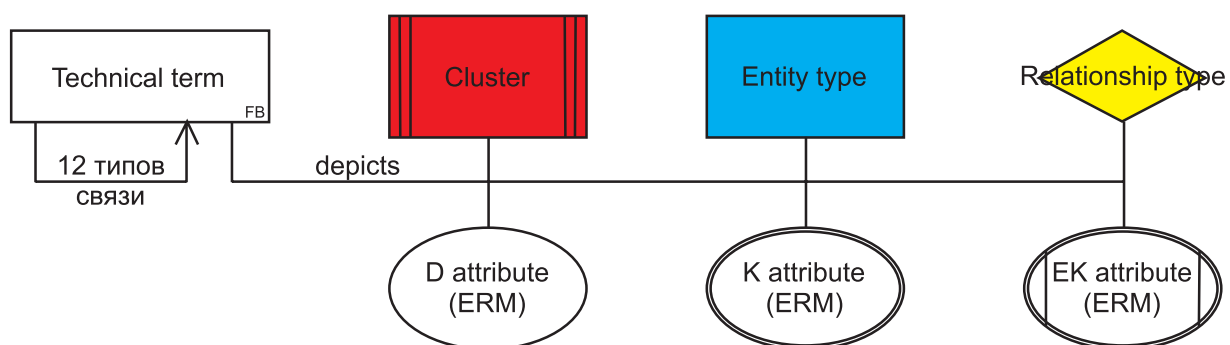


Рис. 6.4.8 Объекты и связи в модели типа Technical terms model  
(Модель технических терминов)

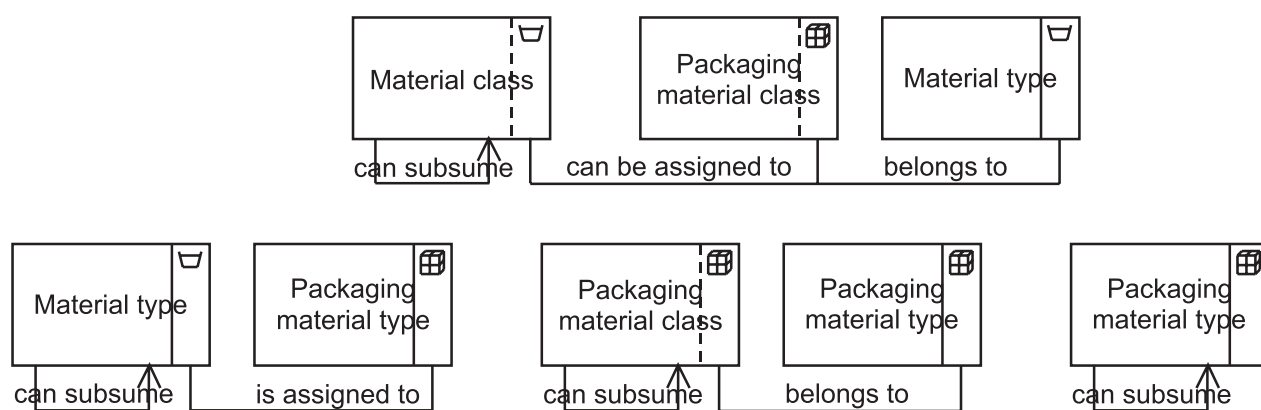


Рис.6.4.9 Типы объектов и связей модели типа Material diagram (Диаграмма материалов)

подчинении). Обе диаграммы позволяют ответить на вопрос: “Откуда берутся стоимости?”. Все стоимости могут быть структурированы в соответствии с различными критериями. Например, по производственным затратам такое структурирование приводит к стоимости производственного персонала (зарплата и начисления на нее), стоимости производственных ресурсов (материалов, заготовок, электроэнергии) и т. д. Стоимостные категории могут также определяться в соответствии с наиболее важными операциями, такими, как стоимость закупок, стоимость хранения на складе, стоимость производства, административные издержки, стоимость продаж и т. д.

Модель типа *Information carrier diagram* (Диаграмма носителей информации) используется для моделирования данных в управлении проектами, где описываются входные и выходные данные в форме документов, журналов, ARIS-диаграмм и т. д. В процессном виде моделирования цепочка управления проектом описывается с помощью диаграммы PPC, где данные описываются в “привязанном” кластере. Необходимые рабочие документы, например, файлы, подготовленные с помощью текстовых редакторов, могут быть представлены и активизированы (вызваны) через атрибуты Link 1, Link 2, Link 3.

На этапе спецификации логические структуры данных, разработанные с учетом сформулированных требований, трансформируются в описание, на основе которого могут быть построены конкретные системы БД.

Модели типа *Relations diagram* (Реляционная диаграмма) и *Attribute allocation diagram* (Диаграмма размещения атрибутов) служат для спецификации сущностей, отношений, атрибутов и их взаимосвязей с информационными объектами, вводимыми при формулировке требований.

Модель типа *System attribute* (Системные атрибуты) дает возможность иерархически упорядочить типы сущностей, события, технические термины, функции, информационные носители, организационные единицы и персонал, а также однозначно и в полном соответствии с требованиями к обработке данных специфицировать их. При обычных требованиях к БД различают следующие типы атрибутов: первичные и внешние ключи, описательные и обязательные поля. Для того, чтобы определить типы областей этих объектов данных, используется модель типа *System attribute domain* (Область системного атрибута). В отличие от ERM-атрибутов основное свойство системных атрибутов состоит в представлении и управлении данными, ориентированными на интерфейс. Системные атрибуты объектов имеют два поля значений, которые заполняются соответствующей информацией. Это гарантирует большую гибкость в экспортируемом содержимом. Диаграмма *Область системного атрибута*

определяет объекты системных атрибутов в соответствии с типом данных. Например, она специфицирует тип области (char, int, date, и т. д.) и длину поля. Эта диаграмма используется, главным образом, для обеспечения информацией процесса экспорта данных из ARIS во внешние прикладные системы.

На этапе описания реализации используется модель типа *Table diagram* (Табличная диаграмма), в которой описываются таблицы и поля систем управления БД.

## 6.5. Функциональное моделирование

Путем выполнения БП осуществляется достижение определенной совокупности целей предприятия. В общем случае совокупность целей можно представить в иерархическом виде (построить “дерево целей”), где каждая цель имеет свой вес и критерий (количественный или качественный) достижимости. Стратегические цели в ходе анализа разбиваются (детализируются) на совокупность частных целей (подцелей). Такое разбиение удобно осуществлять через *декомпозицию* БП, направленных на достижение стратегических целей.

При системном анализе проблем предприятия особое внимание обращают на цели, т. к. проблема – это несоответствие или противоречие между сложившимся и целевым (желаемым) состоянием предприятия. Некорректность, неточность и неопределенность постановки целей способны не только усилить неопределенность самих проблем, но и вместо одной проблемы породить их комплекс, усложнить их анализ, потребовать увеличения объема необходимых ресурсов и т. д.

Целенаправленность предполагает осознанное движение к четкой и ясной цели, несмотря на все препятствия и даже вопреки им. Под *целью* обычно понимают некоторый идеальный, заранее заданный результат, которого следует достичь. *Цель* – это субъективная конструкция, умозрительное построение, зависящее от общего уровня знаний и субъективных качеств аналитика. В бизнесе эта идеальная конструкция характеризуется достаточно однозначно стремлением к максимальной прибыли. Однако часто в экономических процессах и системах цель не имеет четкой формальной выраженности, не поддается исчерпывающему описанию в виде цифр, но ее можно описать в виде схем, зависимостей, диаграмм, связей и т. д.

Понятие цели имеет системообразующий и организующий характер экономической системы. Четко поставленная цель определяет набор БП и структуру предприятия. Она определяет также все аспекты сбора, обработки, систематизации и использования информации.

Кроме описания целей, необходимо указать условия их реализации (ресурсы, сроки, исполнители и т. д.). Общая цель, стоящая перед экономической (производственной) системой или отдельной личностью, может быть подвергнута декомпозиции на составляющие ее подцели различного уровня, вплоть до элементарных. При этом образуется *дерево целей*, которое составляет важную основу современных подходов и способов управления. Важно определить и описать цели как желаемого состояния системы на конкретном этапе ее развития, а также обосновать системный характер цели и ее ресурсного обеспечения, допускающего декомпозицию целей, ресурсов, сроков, исполнителей.

Таким образом, важно иметь инструментарий описания целей предприятия и путей их достижения. Для этого в функциональном виде моделирования имеются соответствующие типы диаграмм.

На рис.6.5.1 для модели типа *Диаграмма целей* (*Objective diagram*) представлены все допустимые объекты-источники, объекты-приемники и связи между ними.

С помощью *диаграммы целей* можно описать цели компании и построить их иерархию. В ARIS *цель* – это описанные результаты деятельности фирмы, которые предполагается достичь, при поддержке способствующих успеху факторов и реализации новых БП. *Факторы успеха*, способствующие достижению сформулированной цели, можно описать путем построения их иерархии и установления соответствия целям. Факторы успеха определяют аспекты деятельности, которые необходимо рассмотреть, чтобы достичь отдельной цели компании. В диаграмме целей фактор успеха должен быть связан с соответствующей целью компании. Этот тип диаграмм связывается с другими диаграммами с помощью функций, так как каждой цели можно поставить в соответствие функцию (БП), которая ведет к достижению цели.

На рис.6.5.2 изображена конкретная диаграмма целей.

Достижение поставленной цели требует предварительного решения комплекса определенных задач. Задачи (функции), как и цели, подчиняются принципам построения и функционирования системы, они могут быть подвергнуты декомпозиции. *Дерево функций* (задач) характеризует задачи, стоящие перед бизнес-системой. Поскольку выполнение задачи приводит к получению результата, образуется логическая цепочка “цель – задачи – результат”. Желательно, чтобы продвижение вдоль звеньев этой цепочки сопровождалось самообучением: цели формулировались и ставились более четко, определенно, конкретно, задачи выявлялись в полном объеме, условия для их выполнения создавались наиболее благоприятные, результаты максимально соответ-

ствовали целям. Всегда полезно осуществить декомпозицию целей, задач и результатов, так как общая цель считается достигнутой, если достигнуты основные частные цели, выполнены основные задачи, а отклонения результата от цели находится в допустимых пределах.

Итак, после построения дерева целей, описываются БП, выполнение которых обеспечивает достижение этих целей. Для статического описания БП в ARIS 5.0 с полным фильтром используются несколько диаграмм. Для описания деятельности предприятия на самом верхнем уровне агрегации используется модель типа *Y-диаграмма*. Левая ветвь Y-диаграммы содержит основные управленческо-административные функции, связанные с планированием и управлением производством, а правая – технико-ориентированные функции производства и реализации продукции. Например, комплексные функции (функции верхнего уровня) производственного предприятия подразделяются на главную функцию (миссию), коммерческую функцию, техническую функцию, финансовую функцию, административно-управленческую функцию. *Главной функцией* или *миссией* предприятия может являться удовлетворение потребностей покупателей и обеспечение прибыли акционеров. Данную функцию реализует процесс деятельности предприятия в целом. *Коммерческая функция* выражается в обеспечении предприятия всеми факторами производства и реализации произведенного продукта. Она реализуется в таких БП, как закупки, продажа и логистика. *Техническая функция* предприятия выражается в создании продукта, удовлетворяющего требованиям потребителей. Данная функция реализуется в таких БП, как исследования (разработки), производство, обеспечение качества продукта и сервисное обслуживание производства. *Финансовая функция* реализуется в учете и финансировании деятельности предприятия. *Административно-управленческая функция* предприятия выражается в организации эффективного управления предприятием и находит отражение в следующих БП: планирование (бюджетирование), управление персоналом, установление связи с общественностью, административно-хозяйственная деятельность, информирование руководства и др.

Основными критериями классификации БП являются степень их важности и характер протекания. Степень важности оценивается по влиянию на выполнение миссии (главной функции предприятия). Характер протекания отражается во взаимодействии или влиянии на другие БП.

По степени важности все БП на предприятии подразделяются на основные и неосновные. Основные БП оказывают существенное и непосредственное влияние на выполнение миссии предприятия. Неосновные БП имеют

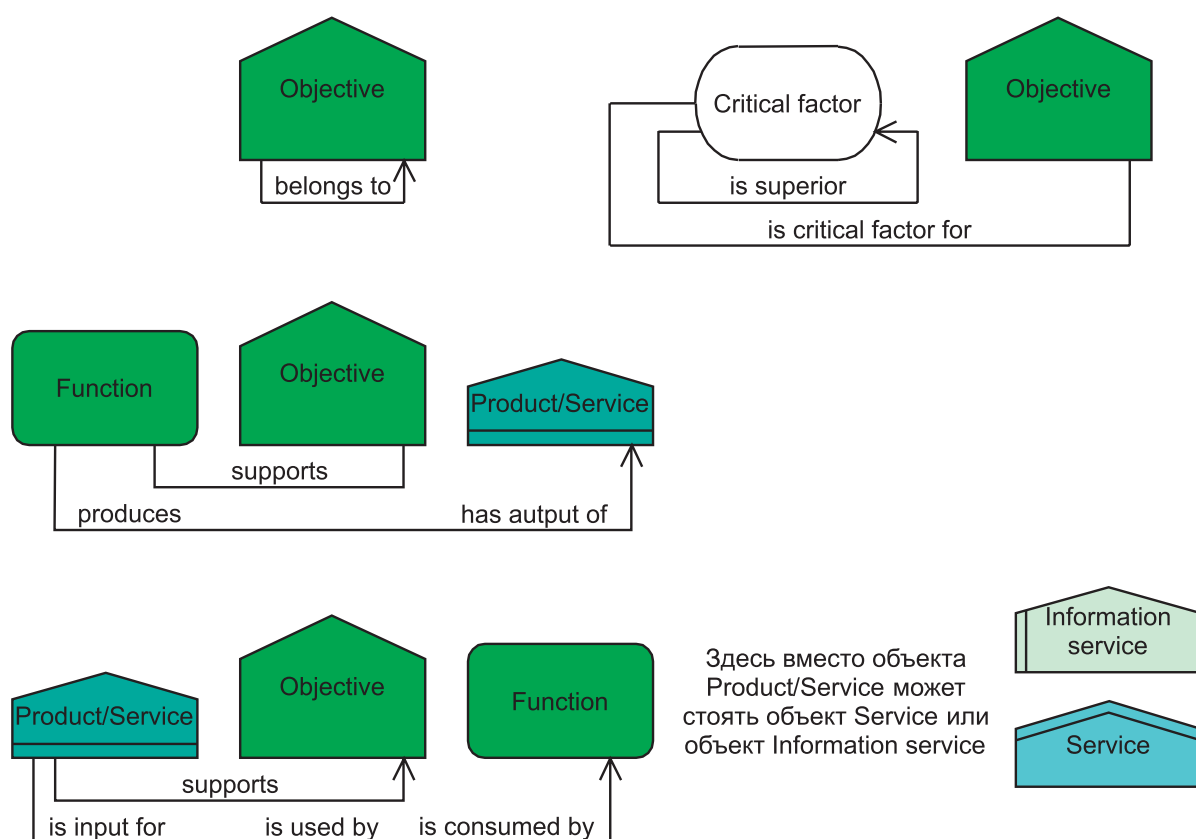


Рис. 6.5.1 Объекты и связи в моделях типа Objective diagram (Диаграмма целей)

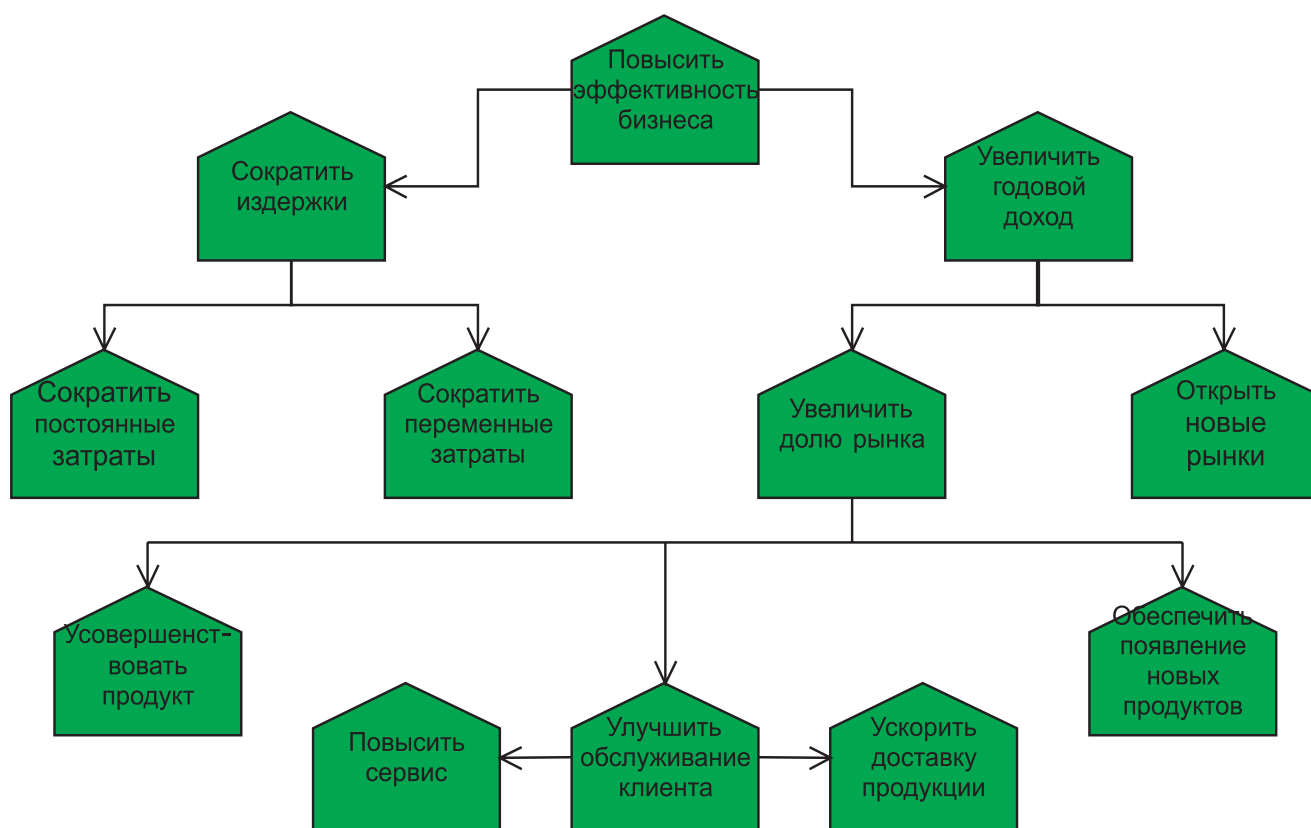


Рис. 6.5.2 Пример диаграммы целей



лишь опосредованное и несущественное влияние.

Основные БП на производственно-сбытовом предприятии – закупки, производство, маркетинг, сбыт и т. д.

*Закупки* – обеспечение производства необходимыми материально-техническими ресурсами при минимизации затрат на их приобретение.

*Производство* – разработка и создание продукта, удовлетворяющего требованиям рынка при минимизации затрат.

*Сбыт* – реализация произведенного на предприятии продукта при максимизации доли на рынке и цен, и минимизации затрат на реализацию.

*Планирование (бюджетирование)* – обеспечение пропорционального, сбалансированного развития предприятия в целом и всех его подразделений в краткосрочной, среднесрочной и долгосрочной перспективе при максимизации результата и минимизации затрат.

*Обеспечение качества* – создание условий эффективного протекания всех БП, создание требований к продукту, соответствующих потребностям рынка, и контроль за их соблюдением.

*Информирование руководства* – создание информационных условий для эффективного принятия руководством предприятия управленческих решений.

К неосновным БП на предприятии относятся логистика, учет, управление персоналом, сервисное обслуживание производства, работа с общественностью, административно-хозяйственная деятельность, обеспечение безопасности и др.

Вопрос о том, что из перечисленной деятельности предприятия описывать в Y-диаграмме, а что в функциональном дереве или другой диаграмме, зависит от целей моделирования и решается ARIS-аналитиком вместе со специалистами предприятия.

Допустимые типы моделей функционального вида в ARIS 5.0 с Fullfilter представлены в приложении 1. Эти типы тесно связаны друг с другом. В функциональных моделях используются только такие средства из других видов моделей, которые отражают связи между функциями. Например, отображение взаимосвязей между данными и функциями дает возможность описать процесс преобразования, выполняемого функцией, через входные и выходные данные для этой функции. Из всех типов моделей дерево функций является наиболее используемым.

Модель типа *Дерево функций (Function tree)* предназначена для представления иерархической структуры функций. Функциональные деревья относятся к одному из семи типов ARIS-моделей с полным методологическим фильтром. Функциональные деревья отобра-

жают статические связи между функциями и позволяют ответить, например, на следующие вопросы: “Зависят ли функции друг от друга? Какая из функций важнее, приоритетнее? Существуют ли между функциями “родительские” отношения?”

*Функция* – это задача, операция или действие, которое выполняется над объектом для достижения одной или нескольких целей предприятия. Каждая функция должна быть названа в форме *глагол + существительное*, т. е. название функции должно содержать действие и объект, на которое это действие направлено. В полном названии (атрибут Full name) можно указать особые условия, ограничения или другую информацию, способствующую однозначному пониманию назначения этой функции разными людьми. Поскольку *функция* – это предметно-ориентированное задание или действие, выполняемое над объектом, в результате которого достигается одна или несколько целей, стоящих перед компанией, то и в названии функции указывается входной объект и действие над ним. В кратком названии функции можно употреблять отглагольные существительные, например, “Производство”, “Реализация” и т. д.

Функции могут быть описаны с различными уровнями детализации. На самом верхнем уровне описываются наиболее сложные функции, представляющие собой отдельный БП или последовательности БП. Последовательная детализация функций образует иерархическую структуру их описаний. Для более содержательного описания отдельного уровня иерархии могут быть использованы также синонимы термина функция, например, *транзакция, процесс, действие, работа, операция* и т. д.

Разделение функций на элементы может происходить на нескольких иерархических уровнях. Базовые функции представляют самый нижний уровень в семантическом дереве функций. *Базовая функция* – это функция, которую с целью анализа БП уже нет смысла разделять на составляющие.

Функция может быть описана с различным уровнем детализации. Например, комплексная функция разбивается на функции, которые делятся на подфункции и т. д. до элементарных (далее неделимых) функций. На рис.6.5.3 комплексная функция “Производство изделия” разбита на функции, подфункции и элементарные функции.

Декомпозиции подвергаются и процессы, и функции, когда последние разбиваются на подфункции. *Декомпозиция функций* – способ построения дерева функций при функциональном моделировании организации. *Дерево функций* – многоуровневое представление функций на основе выбранного способа декомпозиции (деления на составные части). *Способ декомпозиции* –



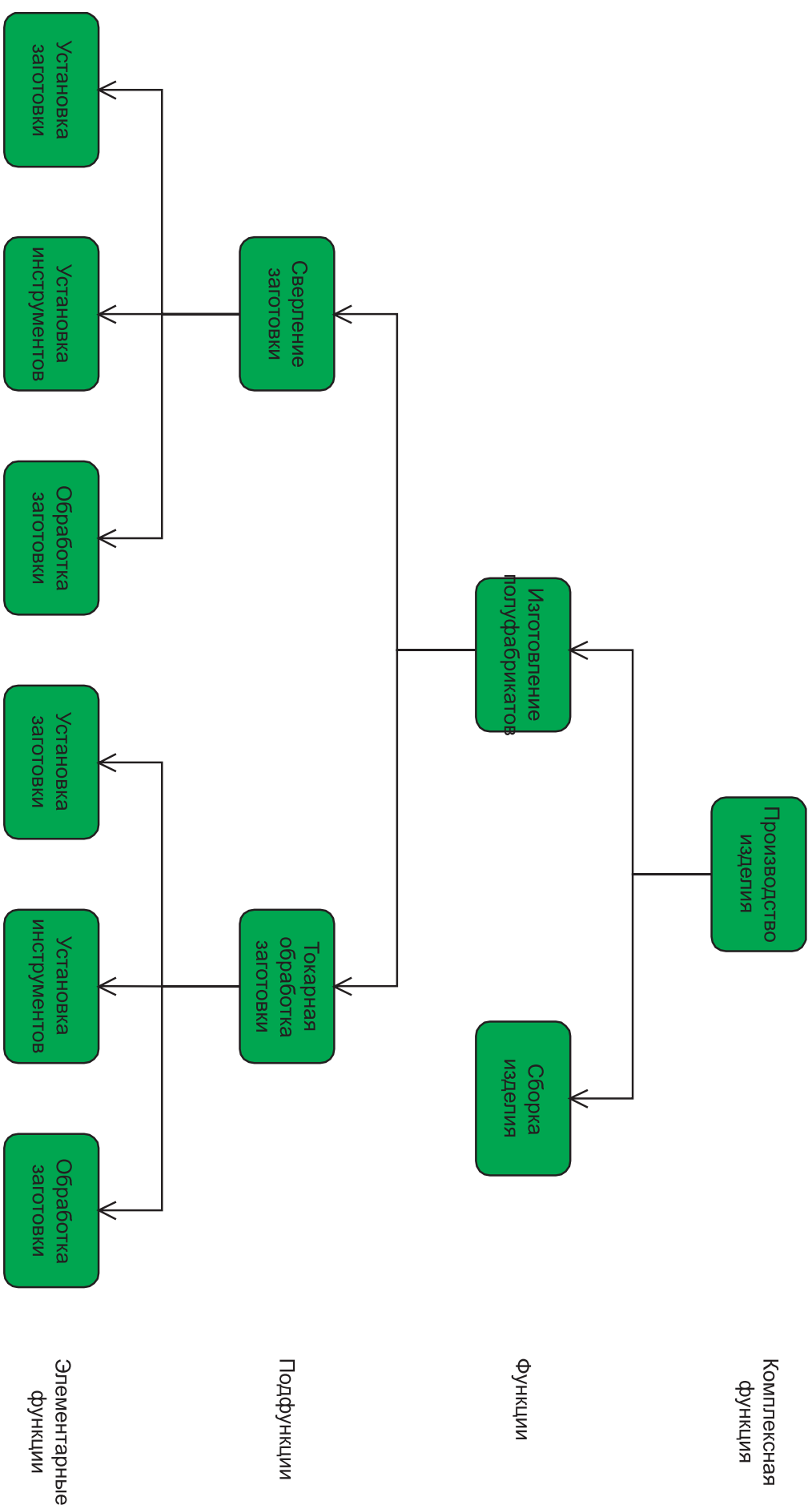


Рис. 6.5.3 Пример иерархической декомпозиции функций

это правило, на основании которого исходная функция по одному или нескольким признакам разделяется на обеспечивающие ее подфункции. Результатом декомпозиции является упорядоченная иерархическая совокупность функций, изображаемая в виде однокоренного графа – дерева функций. Признаками окончания декомпозиции могут быть достижение требуемого уровня детализации функции или появление элементарной функции, т.е. не подлежащей дальнейшему делению.

Число подфункций нижнего уровня, обеспечивающих выполнение одной функции верхнего уровня, обычно составляет 5 – 7. При большем числе выявленных подфункций целесообразно увеличить число функций предыдущего уровня. Число уровней в дереве функций, исходя из требований обозримости и наглядности, обычно также не превышает 5 – 7.

Каждый последующий уровень уменьшает степень общности (абстрактности) функции (БП) и повышает степень детализации описания функции. Подфункции, образующие исходную функцию, называются *детализирующими* или *обеспечивающими*. Каждая обеспечивающая функция должна отличаться от других подфункций и быть необходимой для выполнения исходной.

Так как выполнение каждой функции обеспечивает достижение определенной цели, *“дерево функций”* представляет собой функциональное отражение *“дерева целей”* предприятия. Так как цели связаны с показателями деятельности предприятия, то можно построить *“дерево показателей”*. Эти показатели затем образуют *систему показателей оценки эффективности* выполнения БП. Как правило, владельцы БП контролируют свои БП с помощью данной системы показателей. Наиболее общими показателями оценки эффективности БП являются: количество производимой продукции заданного качества, количество потребителей продукции, количество типовых операций, которые необходимо выполнить при производстве продукции за определенный интервал времени, стоимость издержек производства продукции, длительность выполнения типовых операций, капиталовложения в производство продукции.

Функции объединяются в функциональное дерево в соответствии с различными критериями. Наиболее часто для этих целей используются такие критерии, как обработка одного и того же объекта (*объектно-ориентированный*), принадлежность одному и тому же процессу (*процессно-ориентированный*), выполнение одинаковых операций (*операционно-ориентированный*).

Например, функция *обработать заказ на продукцию* детализируется на функции *создать заказ на продукцию, скорректировать заказ на продукцию, отме-*

*нить заказ на продукцию, обслужить заказ на продукцию, подтвердить заказ на продукцию и управлять заказом на продукцию*. Эти функции описывают различные операции (*создать, скорректировать, отменить* и т. д.), которые выполняются над одним и тем же объектом, в данном случае над объектом *заказ на продукцию*. Таким образом, эта декомпозиция является объектно-ориентированной.

В отличие от объектно-ориентированного разбиения критерием при процессно-ориентированной детализации служит принадлежность функций одному БП, т. е. в названиях функций могут быть различные операции, которые выполняются над различными объектами.

При операционно-ориентированном подходе функция верхнего уровня декомпозируется на подфункции, каждая из которых выполняет ту же операцию, но с различными объектами. Функции могут принадлежать различным процессам и привлекаться к обработке различных объектов. Однако выполняемый ими тип операции над отдельными объектами всегда один и тот же.

В качестве примеров на рис.6.5.4–6.5.6 изображены “веточки” объектно-, процессно- и операционно-ориентированных деревьев. На рис.6.5.7 изображено дерево, в котором каждая ветвь имеет свой тип ориентации.

При *процессной декомпозиции функции* “на один шаг” необходимо осуществить такое разбиение на подфункции, чтобы *функциональный поток*, построенный на их основе, обеспечивал выполнение БП, имя которого – делимая функция. Важно проверить необходимость и достаточность разбиения, т. е. необходимость наличия каждой подфункции (без нее исходная функция не выполнима) и достаточность множества подчиненных подфункций (при наличии и выполнении всех подфункций делимая функция будет реализована).

Еще раз обратим внимание на то, что при операционно-ориентированном разбиении функций в каждой из них операция неизменна, а объект, над которым она выполняется, является переменным. На рис.6.5.8 приведены различные операционно-ориентированные ветви функций, выполняемые в учебных заведениях. При объектно-ориентированном разбиении функций, в каждой из них объект неизменный, а операции над ним меняются. На рис.6.5.9 приведены различные объектно-ориентированные ветви функций, выполняемые в учебных заведениях. При процессно-ориентированном разбиении функции общим для всех функций является только то, что они участники одного процесса. На рис.8.5.2 (смотрите 5-й раздел 8-й главы) представлено процессно-ориентированное дерево функции “Прием на работу преподавателей”. На последнем яру-

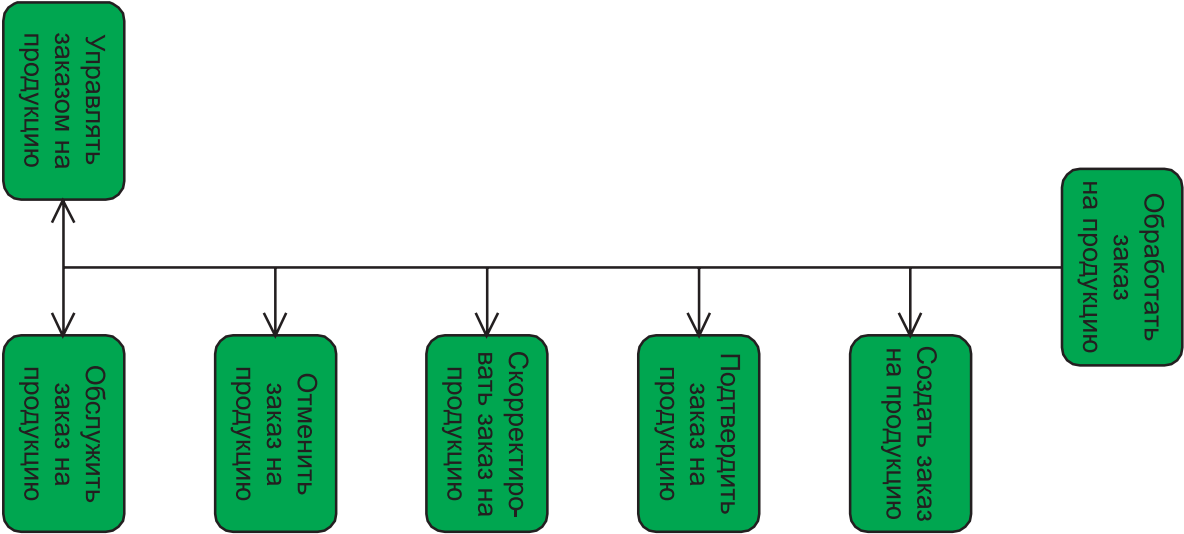


Рис. 6.5.4 Объектно-ориентированное функциональное дерево

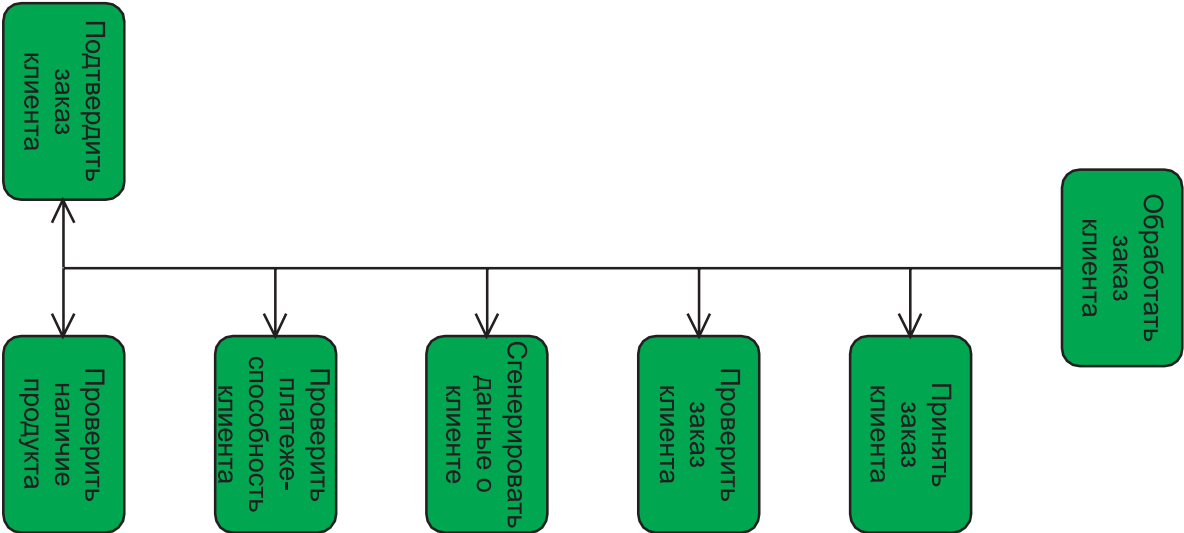


Рис. 6.5.5 Процессно-ориентированное функциональное дерево

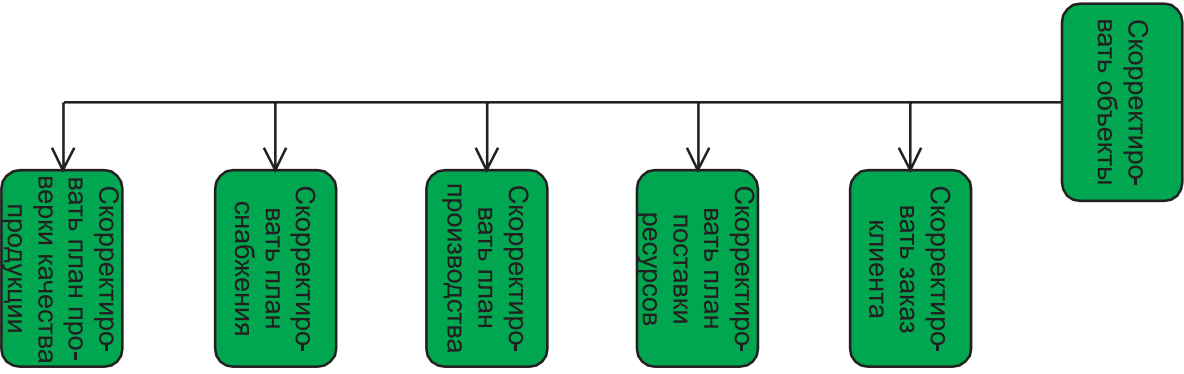


Рис. 6.5.6 Операционно-ориентированное функциональное дерево

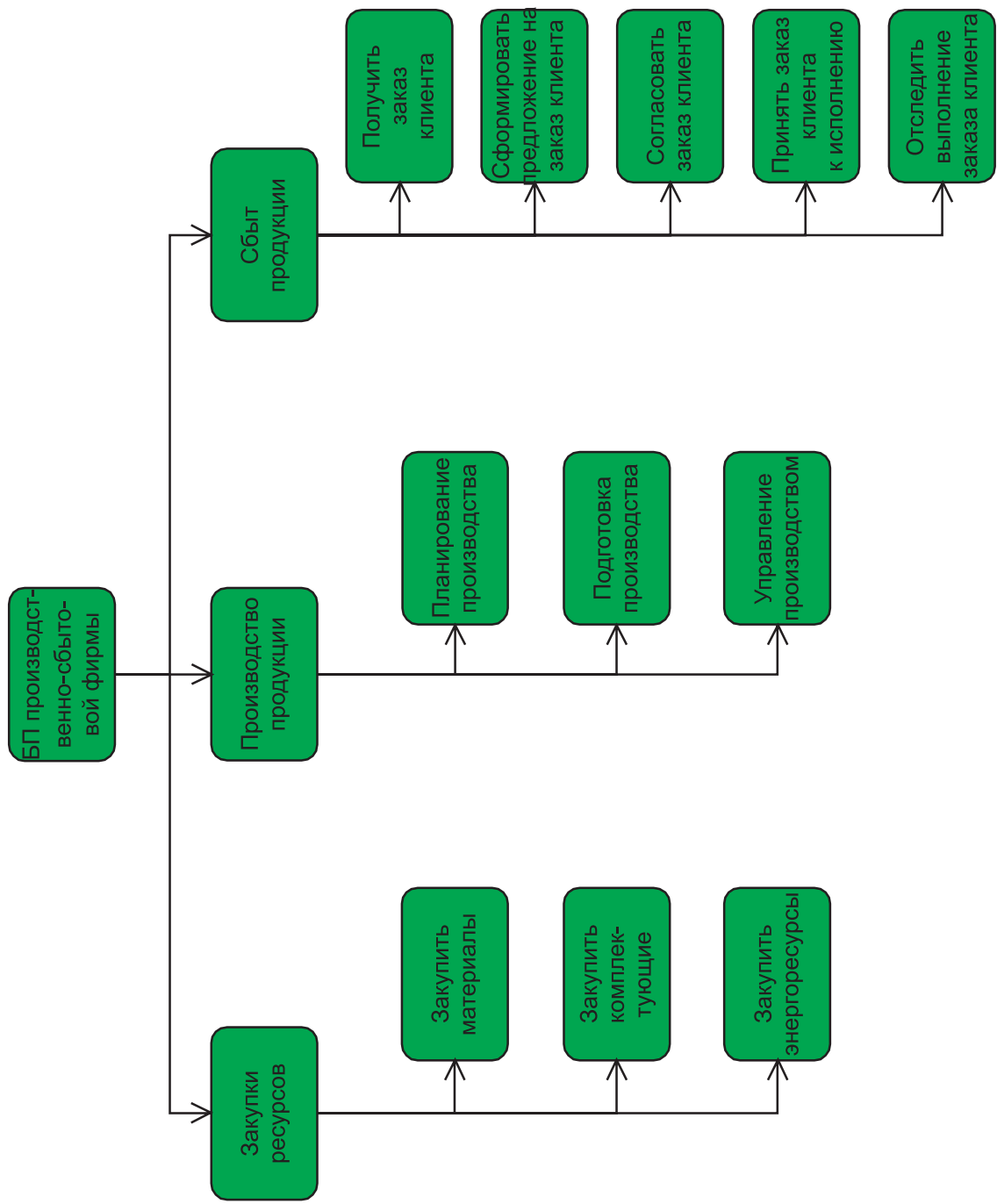


Рис. 6.5.7 Дерево функций бизнес-процесса производственно-сбытовой фирмы

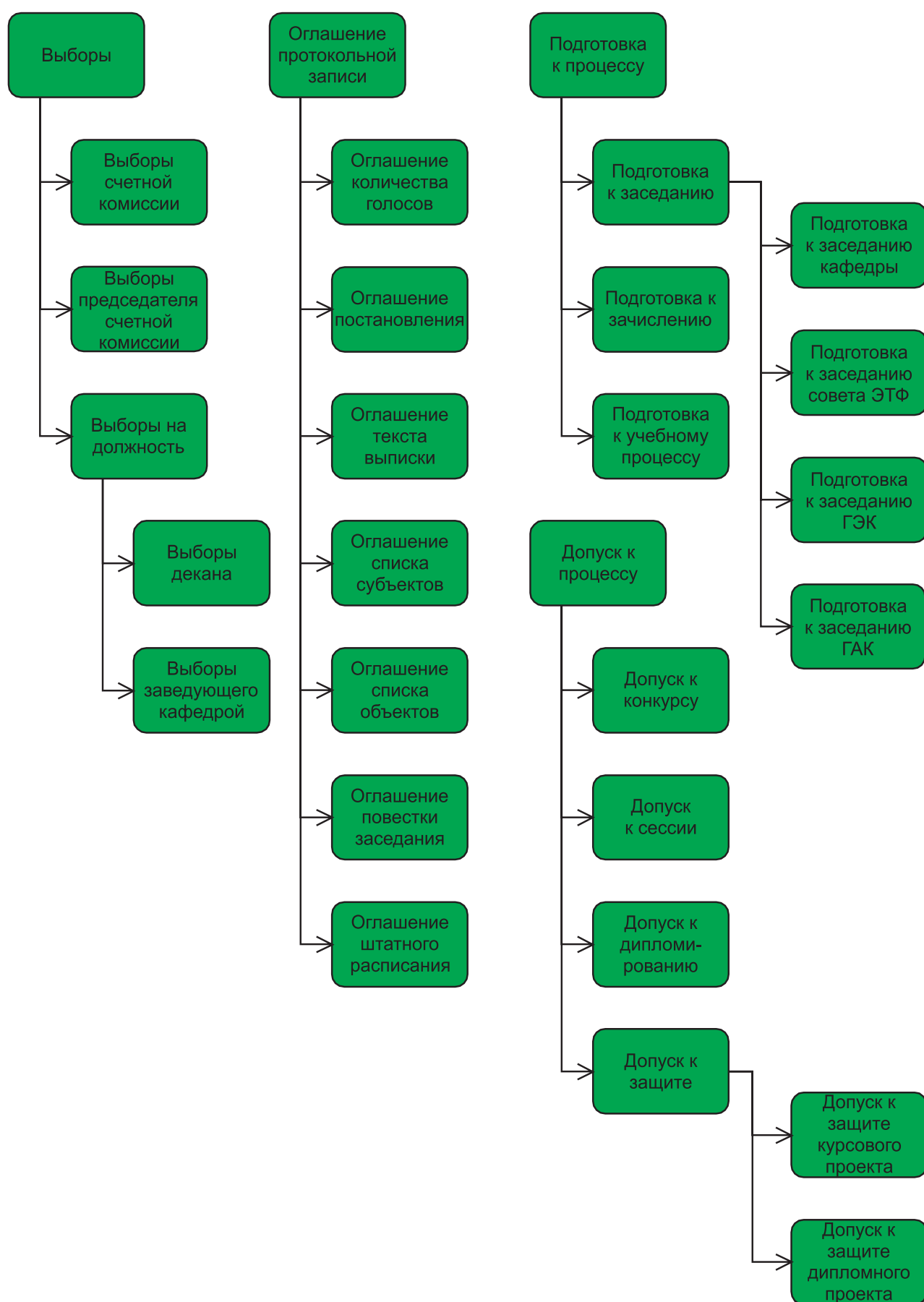


Рис. 6.5.8 Операционно-ориентированные разбиения функций

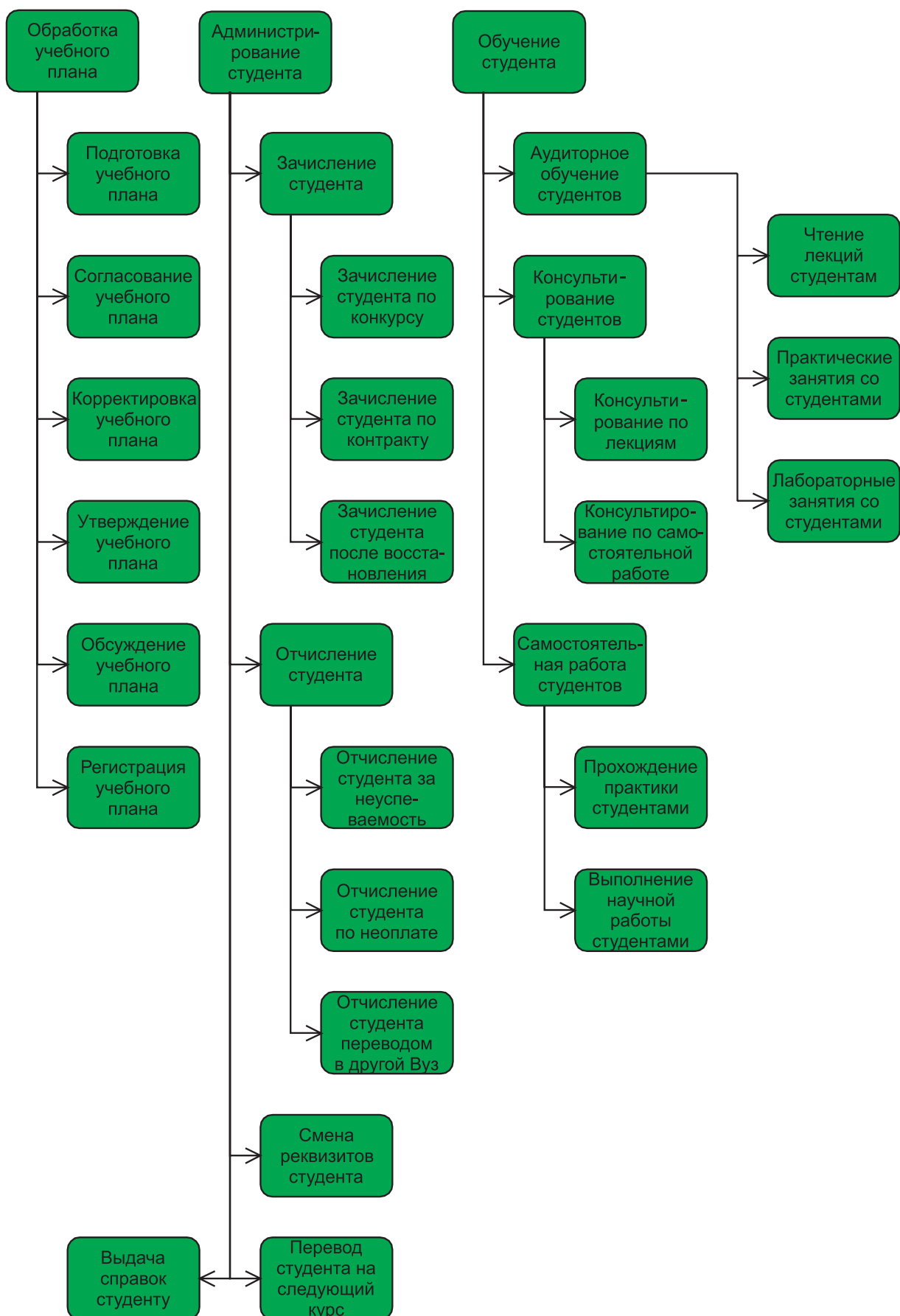


Рис. 6.5.9 Объектно-ориентированное разбиение функций



се (уровне) этого дерева представлены элементарные функции. Процессные модели, построенные на основе этого дерева, представлены в восьмой главе.

При реорганизации БП анализируются количественные характеристики выполняемых функций, например, число запросов, обрабатываемых за день, или совокупное время работы функции, которое формируется из отдельных временных элементов (время настройки, время обработки и время ожидания). ARIS сохраняет эту информацию в атрибутах объекта типа функция.

На этапе спецификации при функциональном моделировании используется *Диаграмма типа прикладной системы*, которая включает спецификацию прикладной системы (ПС) и типов модулей, модульную структуру ПС, прорисовку отдельных шагов-транзакций, а также определение входных и выходных графических интерфейсов. Эта информация предоставляется в виде списков и экранов. На уровне спецификации проекта в рамках функциональной модели необходимо ответить на следующие ключевые вопросы: «Какие типы ПС и модулей поддерживают функции? Какова структура этих ПС и модулей? Какие списки и экраны интерфейсы потребуются для выполнения функции? Какие списки могут быть созданы с помощью ПС данного типа или модуля данного типа и какие экраны поддерживают ПС и модули? Какая технологическая база имеется в распоряжении для реализации данного типа ПС (операционная система, интерфейс пользователя или система управления БД)? Как соотносится с целями компании определенный тип ПС?»

На этапе описания реализации используется *Диаграмма прикладной системы*. С ее помощью можно связать конкретные ПС и модули с типами ПС и типами модулей, как это описано на этапе спецификации. Имеются в виду все копии одного типа ПС, которые могут быть, например, идентифицированы уникальным номером лицензии.

*Прикладная система (модуль)* – это единственный экземпляр типа ПС (типа модуля), который может быть уникально идентифицирован.

Диаграмма прикладной системы отражает фактическую модульную структуру ПС. Можно определить техническую (физическую) реализацию ПС как отдельных программных файлов. Диаграммы ПС наглядно показывают, какие типы программных элементов необходимы для реализации данного типа ПС и типа модуля. *Программный элемент* – это отдельный программный файл, расположенный на носителе данных (например, файл с расширением .exe или .com) и приобретаемый вместе с лицензией. *Типы программных элементов* – это совокупность программных элементов, базирующихся на одной и той же технологии.

## 6.6. Процессный вид моделирования

Взаимосвязь между объектами моделей данных, организационных моделей и функциональных моделей осуществляется с помощью управляющего (процессного) вида моделирования. В зависимости от используемого методологического фильтра в этом виде допустимы от нескольких (пять для простейшего фильтра) типов моделей до нескольких десятков типов моделей. От фильтра зависит также количество допустимых объектов и связей между ними, а также их атрибутов. Например, в модели типа Value-added chain diagram с простейшим фильтром используется пять объектов и три типа связи, а с полным фильтром уже двенадцать объектов и семнадцать типов связи. Аналогичная пропорция распространяется и на атрибуты.

Учитывая большое разнообразие доступных типов моделей процессного вида (около шестидесяти) и ограниченный объем книги, опишем с разной степенью подробности только основные типы моделей процессного вида в ARIS 5.0 с полным методологическим фильтром. Эти типы моделей чаще всего встречаются в литературе по ARIS-моделированию. Из-за большого объема материала в этом разделе используется внутреннее разбиение на подразделы.

Моделирование как создаваемой (проектируемой), так и существующей системы и ее процессов начинается с описания требований к устройству (статические аспекты) и поведению (динамические аспекты) этой системы. Большая часть типов моделей в ARIS 5.0 предназначена для описания требований. Поэтому основная часть этого раздела посвящена изложению некоторых моделей первого этапа моделирования, т. е. фазе анализа требований.

### Модель типа Communications diagram

С модели типа *Communications diagram*, название которой можно перевести как *Коммуникационная диаграмма* или *Диаграмма взаимодействий*, можно начинать построение процессных цепочек, т. к. на основе их анализа удобно строить, например, диаграммы типа eEPC, PCD, PPC. Действительно, в этой диаграмме изображаются организационные единицы (ОЕ) или их типы, а также взаимодействия между ними. Установив последовательность и логику этого взаимодействия, можно получить процессную цепочку, т.е. диаграмму eEPC, PPC и т. д. *Диаграмма коммуникаций* описывает БП и действия, которые выполняются отдельными ОЕ или несколькими ОЕ во взаимодействии друг с другом. Объекты типа *Взаимодействие* можно описать иерархически.

В качестве примера построения, анализа и усовер-

шенствования *Диаграмм взаимодействия* рассмотрим БП “Заключение договора”, который протекает по следующему сценарию. Перед подписанием договора готовится проект договора (ПД), который подвергается всесторонней экспертизе. Менеджер договора (МД) готовит ПД, анализирует замечания экспертов, исправляет ПД, согласовывает ПД, подписывает договор у соответствующих сторон, регистрирует договор и передает его адресатам с необходимыми копиями. Техническую экспертизу ПД проводит технический эксперт. Он готовит пояснительную записку (ПЗ) с результатами экспертизы и передает ее МД для уточнения (улучшения) ПД. Аналогичную работу проводят экономист, юрист и информатик, осуществляя экономическую, финансовую и юридическую экспертизу, а также экспертизу на информационную безопасность.

На рис.6.6.1 описано последовательное взаимодействие перечисленных ОЕ в описанном БП. Здесь ПД последовательно проходит через всех экспертов, начиная с технического эксперта. На рис.6.6.2 в этом же БП те же ОЕ взаимодействуют параллельно. Действительно, МД передает ПД сразу всем экспертам, которые работают с ним параллельно и независимо друг от друга.

Диаграммы на рис.6.6.1, 6.6.2 описывают как две ОЕ взаимодействуют друг с другом. Здесь описываются также те действия, которые выполняются внутри отдельной ОЕ.

### Модель типа Value-added chain diagram

Для уменьшения сложности описания деятельности предприятия необходимо разработать иерархию моделей БП предприятия, начиная с самого верхнего уровня и до моделей отдельных БП на нижнем уровне. Для описания модели верхнего уровня используется диаграмма типа *Value-added chain diagram (VAD)*, название которой можно перевести как *Модель цепочки добавленного качества (стоимости)*. В общем случае *цепочка добавленного качества (стоимости)* – это согласованный набор видов деятельности предприятия, которые либо создают добавленную стоимость для основной продукции (товаров, услуг) предприятия, либо добавляют новое качество продукции, либо осуществляют основные переделы продукции, либо одновременно изготавливают продукцию с добавлением нового качества и стоимости, начиная от исходных (входных) ресурсов вплоть до готовой продукции (услуги), доставленной конечному потребителю. Поэтому условимся коротко называть эту модель *Добавляющей цепочкой (ДЦ)*.

Основную роль в ДЦ выполняют *функции*, выходом которых являются продукты (услуги) с измененным качеством и добавленной стоимостью. Эта модель описывает, во-первых, иерархию деятельности компании, во-вторых,

последовательность деятельности (комплексных функций) на каждом иерархическом уровне. Представление деятельности компании в нотации ДЦ подчиняется следующим правилам: функции могут размещаться в функциональной последовательности в соответствии с последовательностью переделов, добавлением качества и стоимости; между функциями могут устанавливаться иерархические связи или отношения; функции могут быть разделены на подфункции.

В многоуровневой ДЦ первый уровень, как правило, состоит из одной функции, в названии которой отражена деятельность предприятия, следующая из ее миссии. На втором уровне перечисляются основные виды деятельности, которые необходимы и достаточны для выполнения миссии. Некоторые или все виды деятельности связываются в последовательные цепочки. Например, для производственно-сбытовой фирмы в эту цепочку входят как минимум БП снабжения, БП производства и БП реализации. Один из следующих уровней ДЦ может быть уровнем оперативного управления детализируемого БП. Тогда следующие за ним уровни образуют функции, входящие в текущую оперативную деятельность. Следует стремиться к тому, чтобы число уровней в модели не превышало семи. Иначе могут возникнуть проблемы использования этой модели, например, ее удобное размещение на экране монитора или листе бумаги, а также анализ диаграммы будут затруднены.

В ДЦ функции выстраиваются не только в порядке их выполнения, но и могут быть упорядочены иерархически по аналогии с функциональным деревом. Такое представление всегда включает отношения старшинства и подчиненности, ориентированные на БП. ДЦ позволяет отобразить не только подчиненность и последовательность функций, но и их связи с ОЕ и информационными объектами.

На рис.6.6.3 представлены все допустимые объекты и связи между ними в моделях типа Value-added chain diagram. В таблице 6.6.1 представлены названия всех допустимых типов связи, порядковые номера которых представлены на рис.6.6.3 рядом с соответствующей связью.

На рис.6.6.4, 6.6.5 представлены примеры ДЦ. В этих примерах ДЦ использованы для описания иерархии основных БП организаций и размещения их в определенной последовательности. На более низких уровнях детализации можно представить сложные БП в виде другой ДЦ или в виде процессных цепочек. Описание структуры БП можно дополнить моделями данных и организационными диаграммами. Детальное рассмотрение функций, описанных в ДЦ, можно осуществлять путем применения *процессных цепочек*. Наиболее часто для этого применяются событийно-управляемые последовательности функций.

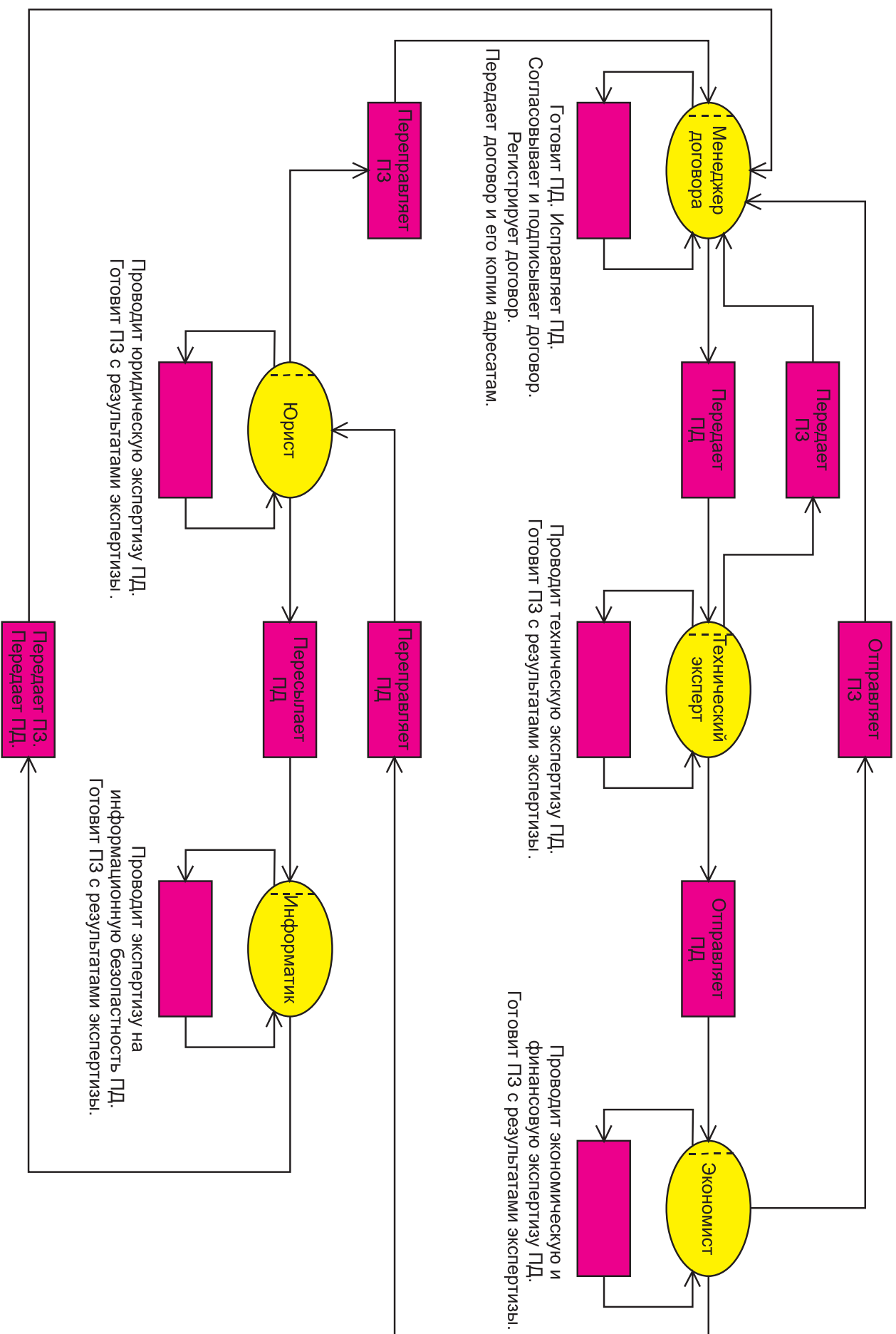


Рис. 6.6.1 Диаграмма последовательного взаимодействия субъектов ответственности БП “Заключение договора”

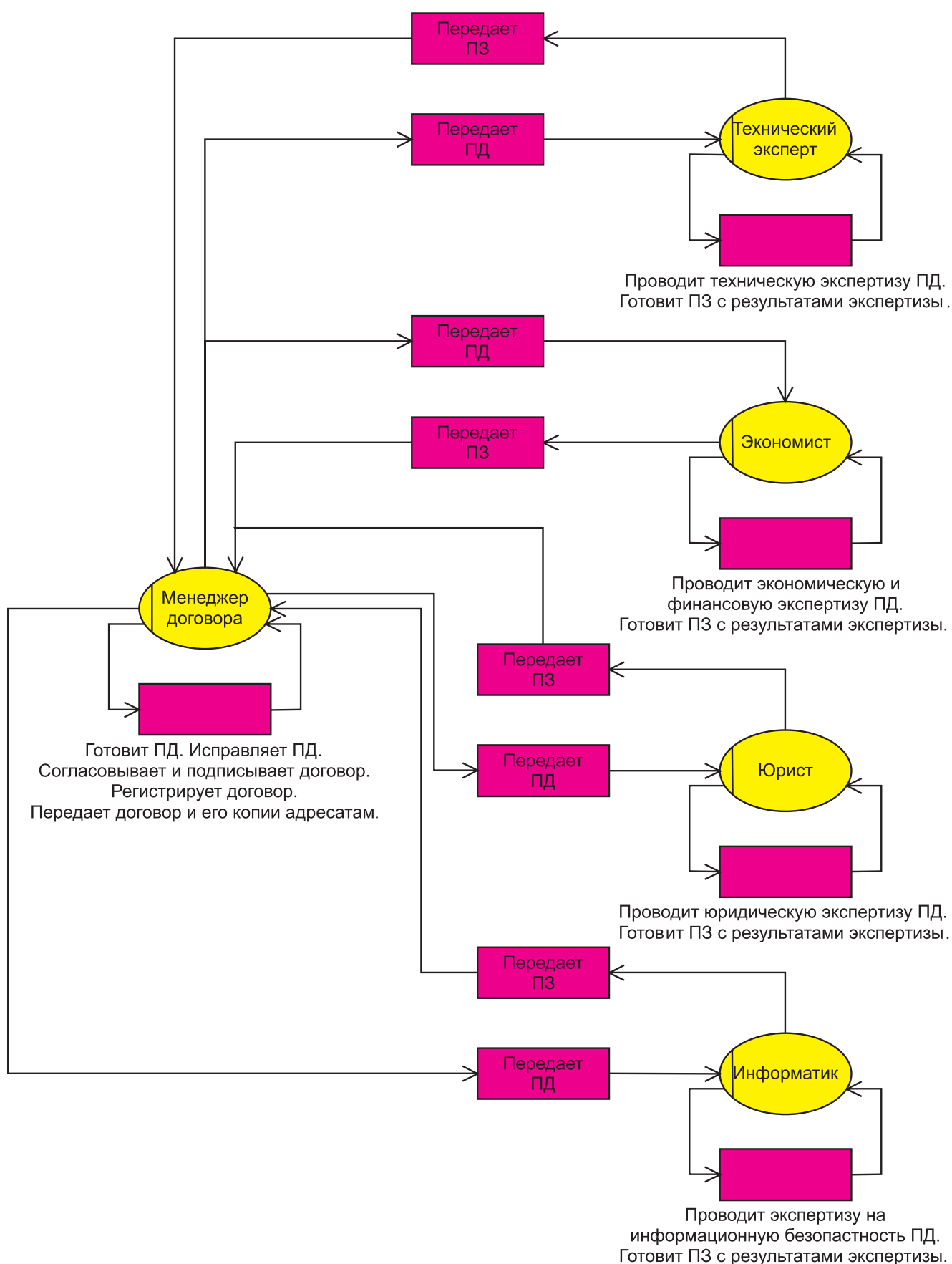


Рис. 6.6.2 Диаграмма параллельного взаимодействия субъектов ответственности в БП "Заключение договора"

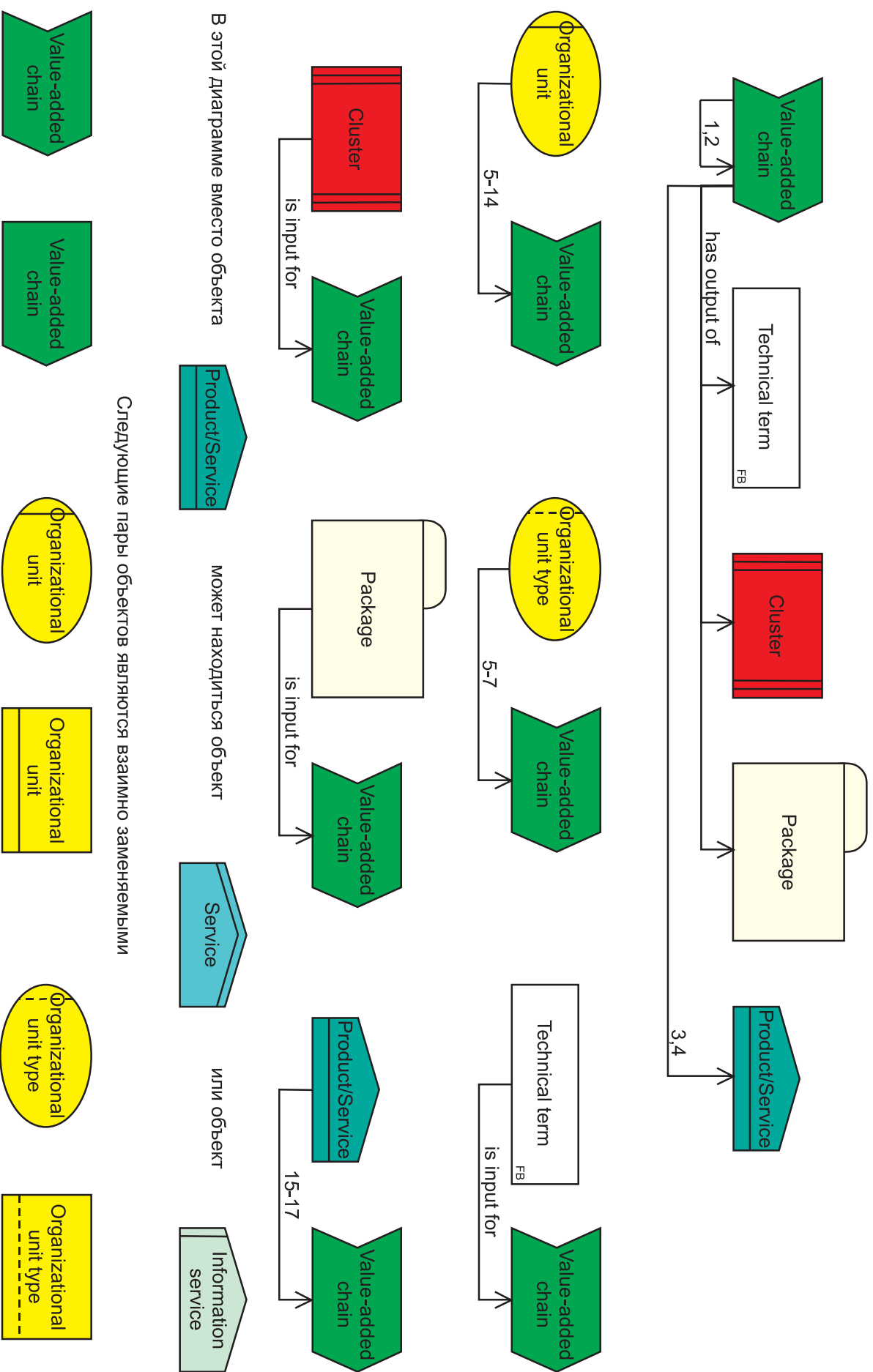


Рис. 6.6.3 Объекты и связи диаграммы типа Value added chain diagram (Диаграмма добавленного качества)

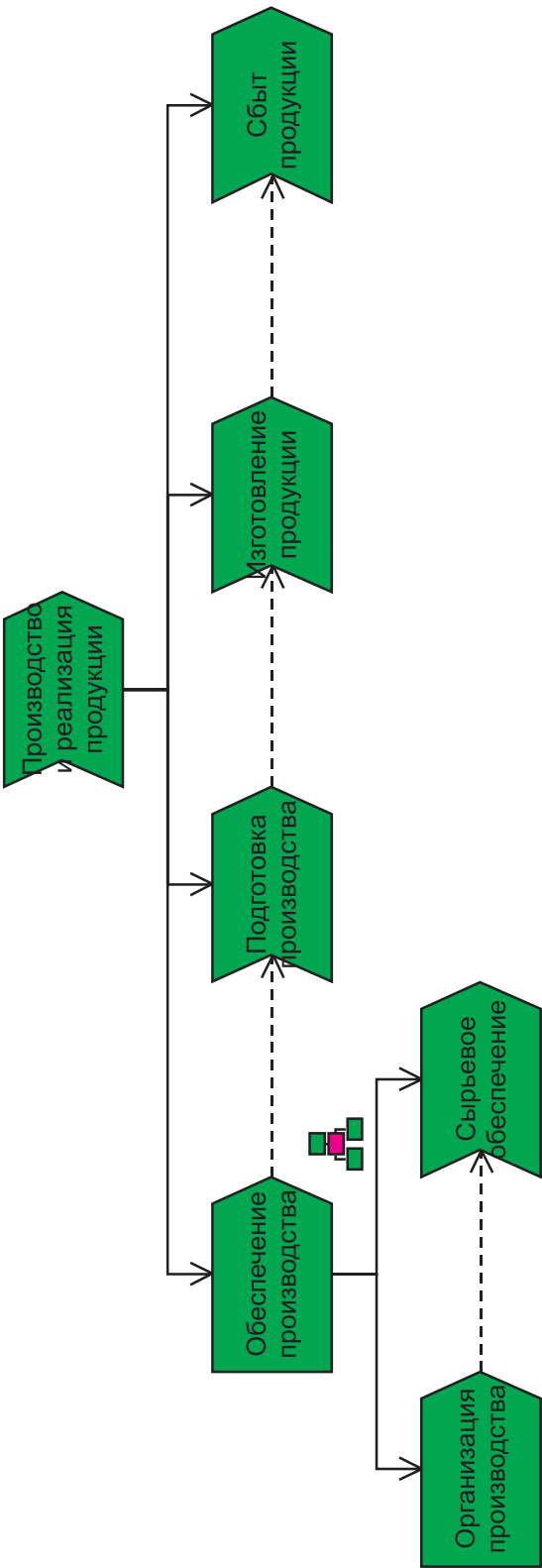


Рис. 6.6.4 Схема взаимодействия основных БП производственно-сбытовой фирмы

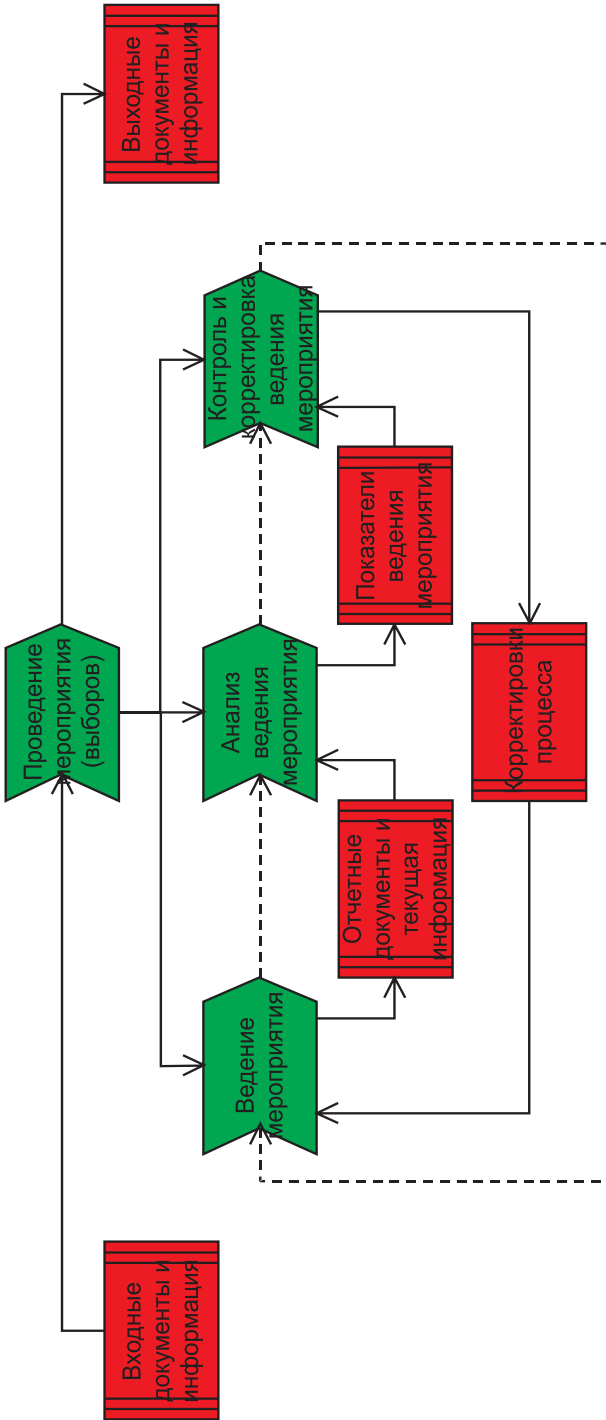


Рис. 6.6.5 Схема непрерывного взаимодействия комплексных функций процесса проведения мероприятия



Таблица 6.6.1

| <i>N</i> | <i>Название типа связи</i>       | <i>Перевод названия связи</i>                       |
|----------|----------------------------------|---|
| 1        | Is process-oriented superior     | является процессно-ориентированным вышестоящим      |
| 2        | Is predecessor of                | предшествует  |
| 3        | Has output of                    | имеет на выходе                                     |
| 4        | Produces                         | производит  |
| 5        | Is technically responsible for   | отвечает за техническую часть                       |
| 6        | Executes                         | выполняет   |
| 7        | Is IT responsible for            | отвечает по ИТ за                                   |
| 8        | Decides on                       | принимает решение по                                |
| 9        | Contributes to                   | способствует при выполнении                         |
| 10       | Must inform about result of      | должен информировать о результатах выполнения       |
| 11       | Must be informed about           | должен быть информирован о выполнении               |
| 12       | Must be informed on cancellation | должен быть информирован о нестандартном завершении |
| 13       | Has consulting role in           | участвует в качестве консультанта                   |
| 14       | Accepts                          | утверждает результат                                |
| 15       | Is input for                     | является входом для                                 |
| 16       | Is used by                       | используется  |
| 17       | Is consumed by                   | потребляется  |

### Модели типа EPCs

Объединение функций и данных описывается в моделях типа EPCs (*Event-Driven Process Chain* – *цепочка процесса, управляемая событиями*), где для каждой функции могут быть определены начальные и конечные события, которые переключают функции (передают управление от одной функции к другой) и являются результатом их выполнения. Заметим, что функциональные потоки в ДЦ статичны. Для получения динамической последовательности функций в функциональный поток необходимо включить *события*, т. е. наборы данных, определяющие состояния и условия, необходимые для выполнения функциональных последовательностей. События могут инициировать начало выполнения последовательности функций (*запускающие события*), ее завершение (*завершающие события*) или изменение порядка выполнения функций в рамках БП (*переключающие события*). Сочетание функции и запускающих её событий образует триггер (управляющий элемент).

В отличие от функций, которые имеют некоторую продолжительность, события происходят мгновенно. События вместе с функциями играют ключевую роль в процессных цепочках. Они запускают (активизируют) функции и являются результатом их выполнения. События описывают состояние объекта и позволяют контролировать БП или влиять на ход его выполнения. Несколько событий связываются с функциями при помощи логических операторов (“и”, “или”, “исключающее или” и др.).

События, например, изменяют состояние информационных объектов модели данных и, следовательно, связы-

вают EPC модели с моделями представления данных. Поскольку информационные объекты и атрибуты описаны в модели данных, *событийная цепочка процессов* является мостиком между моделью данных и функциональной моделью. В названии события должна присутствовать сущность (объект, субъект, понятие и т. д.) и ее изменившееся состояние. Например, события “заказ получен”, “клиент согласен”, “тепло подано”, “цена установлена” указывают на сущность (заказ, клиент, тепло, цена) и их состояние (получен, согласен, подано, установлена).

Возьмем для примера процесс рассмотрения финансовой службой кредитного договора. Запускающим событием здесь является факт возникновения проекта договора в финансовой службе. Завершающими – принятие договора к подписанию и исполнению или отказ от заключения договора. Переключающими событиями – соответствие проекта договора бюджету предприятия, удовлетворительные или неудовлетворительные условия договора и т. д.

С помощью диаграмм EPC процедуры БП представляются как логические последовательности событий. События определяют, какое состояние или отношение будет переключать функцию и какое состояние наступит после ее выполнения. Поэтому модель EPC должна всегда иметь запускающие и завершающие события.

Одно событие может инициировать выполнение одновременно нескольких функций, и, наоборот, функция может быть результатом наступления нескольких событий. Объединения нескольких событий или функций отображаются на диаграмме EPC с помощью соединителей в виде небольшого кружка. Эти соединители не только

отображают графические связи между объектами модели, но и определяют логические связи между ними. Различают два типа связи логических операторов – *связи событий* и *связи функций*. На рис.6.6.6 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), переключающее событие не должно быть связано с функциями операторами OR или XOR! Перечислим допустимые логические операторы.

1. *Соединение переключающих событий*. Если события соединяются оператором AND (и), то выполнение следующей за ними функции может быть начато после того, как произойдут все события (смотрите рис.6.6.6.a). Если события соединяются оператором OR (или), то функция выполняется, если произойдет, по крайней мере, одно событие (смотрите рис.6.6.6.b). Если события соединяются оператором XOR (исключающее или), то функция начинает выполняться после того, как произойдет одно (и только одно) событие (смотрите рис.6.6.6.c).

2. *Соединение сгенерированных событий*. Если функция соединяется с событиями оператором AND, то в результате выполнения этой функции происходят все события (рис.6.6.6.a). Если функция соединяется с событиями оператором OR, то в результате выполнения функции происходит, по крайней мере, одно событие (рис.6.6.6.b). Если функция соединяется с событиями оператором XOR, то в результате выполнения функции происходит максимум одно событие (рис.6.6.6.c).

3. *Соединение функций со сгенерированным событием*. Если функции соединяются друг с другом оператором AND, то следующее за ними событие происходит только после того, как все функции выполнены (рис.6.6.6.d). Если функции соединяются друг с другом оператором OR, то событие произойдет после того, как будет выполнена, по крайней мере, одна функция (рис.6.6.6.e). Если функции соединяются друг с другом оператором XOR, то событие произойдет после того, как будет выполнена одна (и только одна) функция (рис.6.6.6.f).

4. *Соединение функций с переключающим событием*. Если переключающее событие связано с следующими за ним функциями оператором AND, то это событие переключает на выполнение все функции (рис.6.6.6.d). Событие не может принимать решения. Поэтому после события не может стоять оператор OR или XOR, за которым следуют функции.

Для подробного описания деятельности предприятия применяются модели типа eEPC (*extended Eventdriven Process Chain*), т. е. *расширенные событийно-управляе-*

*мые цепочки функций*. В этих моделях отображается, во-первых, последовательность выполнения функций БП, где для каждой функции определены начальные и конечные события, во-вторых, логика выполнения БП, в-третьих, результаты выполнения функций и многое другое.

При оформлении диаграмм функциональных потоков рекомендуется придерживаться следующих ограничений. Во-первых, линейный участок размещать вертикально, т. е. сверху вниз. Во-вторых, в точках разветвления процесса осуществлять горизонтальную компоновку. В-третьих, при наличии свободного места размещать ОЕ и субъекты ответственности справа или слева от соответствующей функции. Если придерживаться перечисленных ограничений, то получится легко читаемый поток и появится возможность использования встроенных в ARIS средств автоматического выравнивания объектов по вертикали и горизонтали.

В процессно-ориентированном описании следует выделять так называемые исключительные ситуации, т. е. условия, когда нарушается нормальное функционирование системы. Выявление и тщательное описание исключительных ситуаций и реакции на них очень важно, но оно может занять много времени. Поэтому особые случаи (ситуации) рекомендуется описывать после завершения основного описания.

В качестве примера на рис.6.6.7, 6.6.8 изображены eEPC-модели БП “Заключение договора”. Их построение основано на диаграммах взаимодействия, которые изображены на рис.6.6.1, 6.6.2.

Диаграмма типа *eEPC с потоком материалов* является расширением диаграммы eEPC. Кроме типов объектов eEPC в этой диаграмме используются следующие типы объектов: *тип материала, тип упаковочного материала, тип операционных ресурсов, операционный ресурс, тип операционного технического обеспечения, операционное техническое обеспечение, тип складского оборудования, складское оборудование, тип транспортной системы, транспортная система*. Тип материала может быть связан с функцией. Если связь осуществляется в виде входящего (выходящего) соединения, то материалы определяются как вход (выход) для функции. Для преобразования материалов требуются *технические ресурсы*. В цепочке процессов их можно связать с объектами типа *функция*. Для описания возможных альтернативных ресурсов используются два типа связи – *требуется* и *требуется альтернативно*. Если для выполнения функции материалы должны быть упакованы, то можно описать *типы упаковочных материалов*, установив взаимосвязь между функцией и необходимыми типами упаковочных материалов.

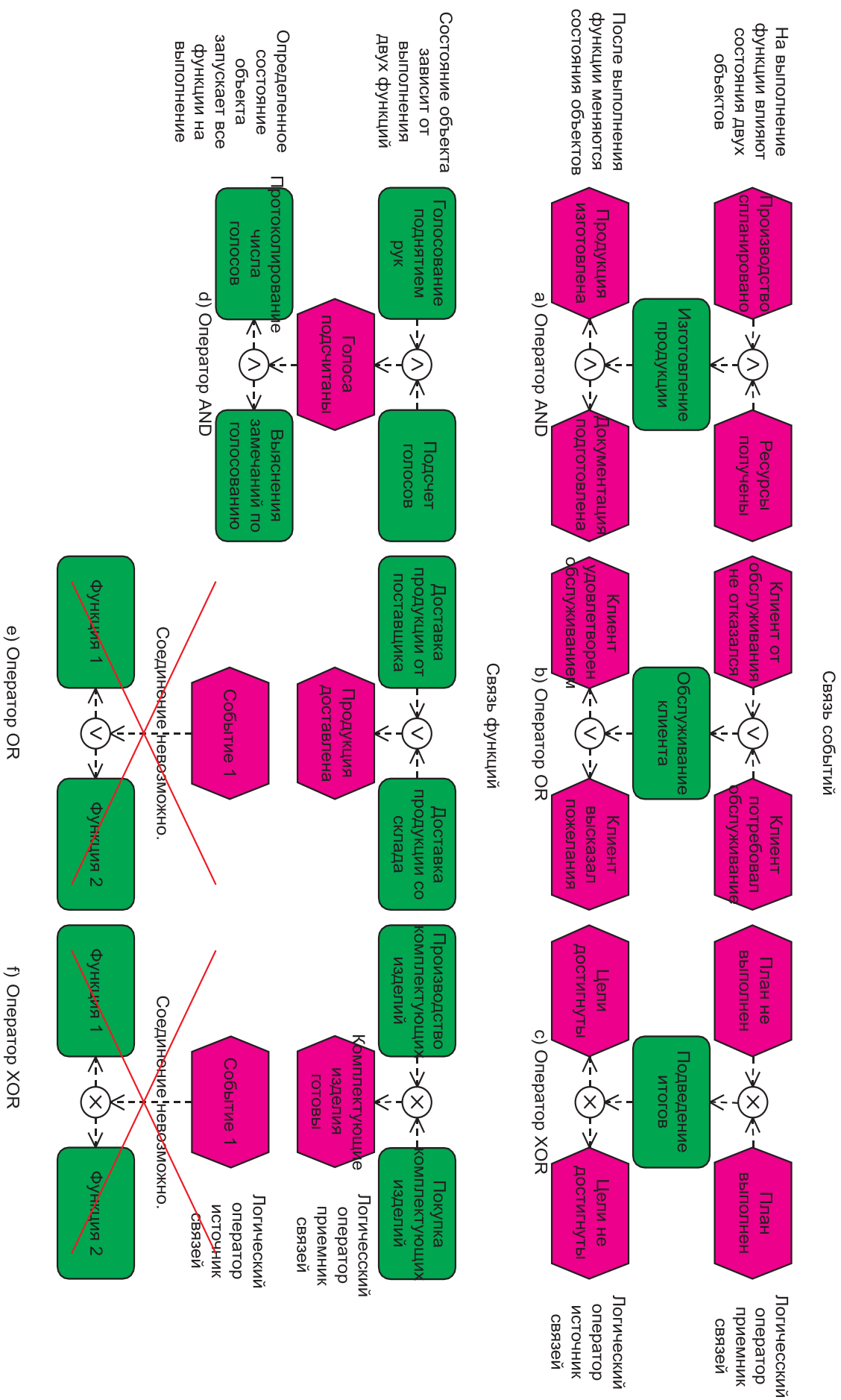


Рис. 6.6.6 Перечень допустимых операторов событий и функций

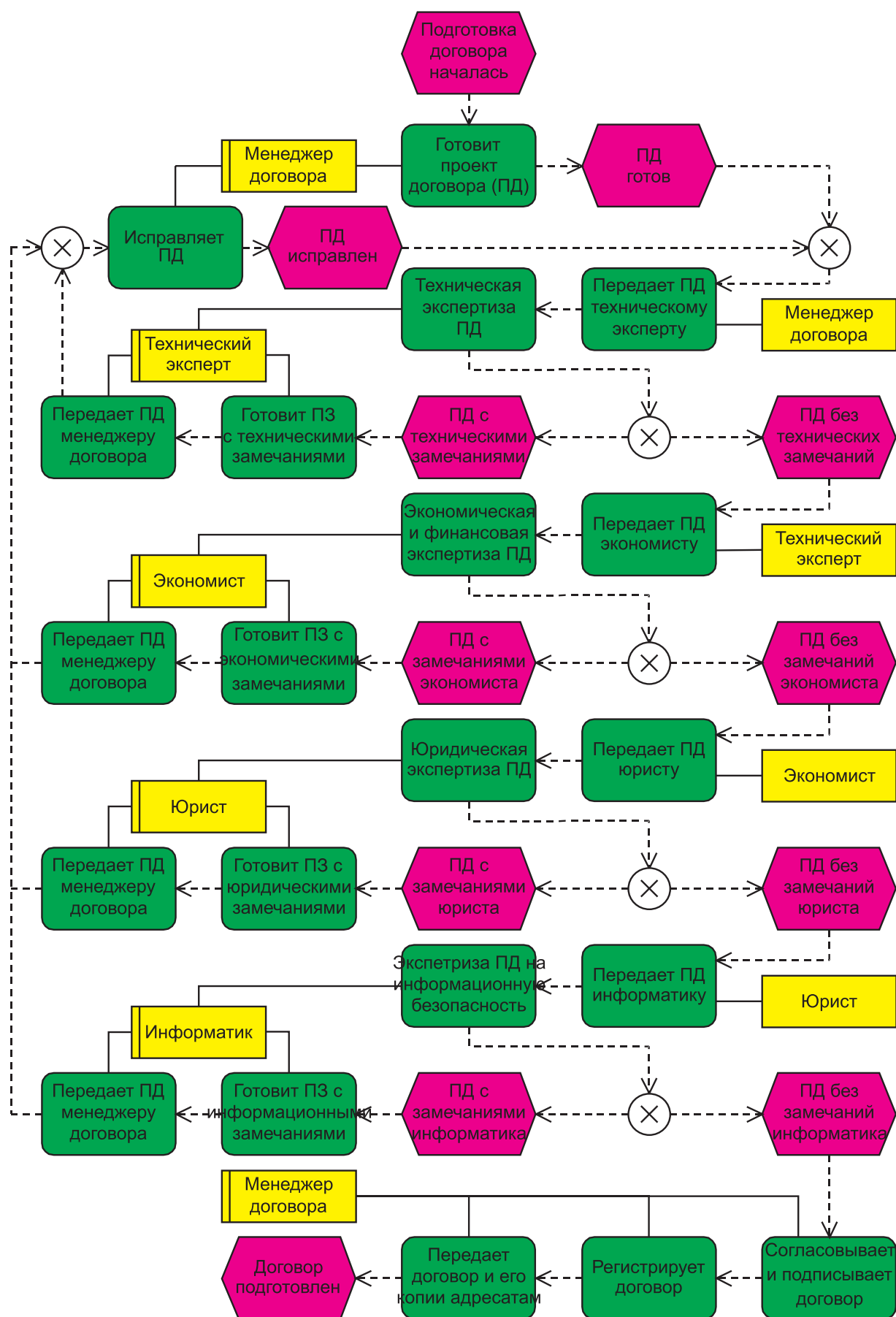


Рис. 6.6.7 Процесс последовательной подготовки договора”

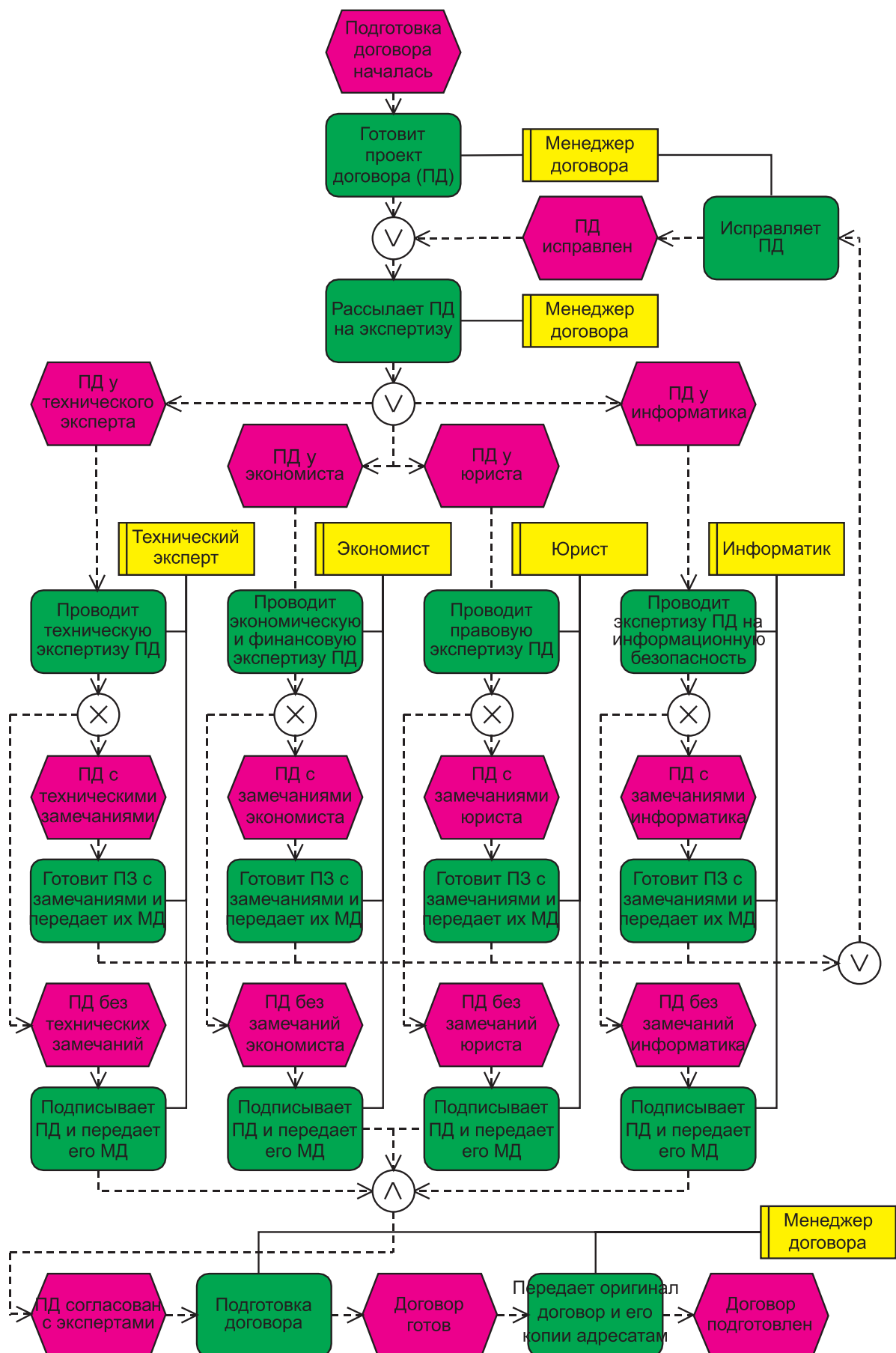


Рис. 6.6.8 Процесс параллельной подготовки договора



Для проведения презентаций перед руководством предприятия применяются, как правило, диаграммы *офисных и производственных (индустриальных) БП*. Они представляют собой особый вид eEPC-модели БП. Модели офисного и производственного процессов строятся с помощью специальных пиктограмм. Они носят “понятийную мысль” и предназначены для более доходчивого понимания модели БП, представленной с помощью eEPC-нотации, и могут быть автоматически получены из нее. Эти диаграммы описывают практически то же самое, что и диаграмма типа eEPC. Но они используют меньше объектов, которые представляются наглядными пиктограммами, максимально соответствующими процессам создания материальных продуктов или нематериальных услуг.

В таблице 6.6.2 сравниваются символы диаграмм промышленного и офисного процессов с символикой eEPC (eEPC с потоком материалов). ARIS автоматически конвертирует символы этих типов диаграмм. На рис.6.6.9 описан один и тот же фрагмент БП диаграммами трех типов – eEPC, промышленного и офисного процессов.

## Другие типы моделей

Модели типа eEPC и PCD объединяют объекты диаграмм всех трех видов моделирования. Эти модели описывают одни и те же вещи. Но в отличие от eEPC на расположение объектов модели PCD накладываются жесткие ограничения. Действительно, диаграмма PCD в целом располагается в вертикальных столбцах таблицы и определенные объекты размещаются в определенных столбцах. События, представляющие объекты модели данных, размещены в первом столбце. Стрелки оканчиваются во втором столбце, который содержит функции процесса. Таким образом, первый и второй столбцы определяют управление событиями. Данные размещаются в третьем столбце, где отображены их связи с отдельными функциями. Таким образом, второй и третий столбцы PCD определяют потоки данных в цепочке процесса. В четвертом столбце размещаются ОЕ из организационной модели, которые ответственны за выполнение отдельных функций в цепочке процесса и т. д.

На рис.6.6.10 показаны возможные уровни детализации процессных цепочек, в которых задействованы модели типа Value-added chain diagram, eEPC и PCD.

С событиями связываются конкретные информационные объекты модели данных, и события описывают факт изменения состояния этих информационных объектов. *Диаграмма событий* позволяет описать событие любой сложности в той или иной диаграмме, например, в eEPC. Для этого с рассматриваемым событием связы-

вается диаграмма событий, в которой детализируется это событие с использованием правил для логических операторов, связывающих события. Поэтому в диаграмме событий используются информационные объекты модели данных, связанные с событиями. Таким образом устанавливается, какое событие определяет изменение состояния данного информационного объекта.

В процессных цепочках (диаграммах типа eEPC и PCD) можно использовать правила для описания сложных логических операторов, которые связывают события и функции. Сложный логический оператор можно связать с *Диаграммой правил*, которая описывает все детали сложного правила, в том числе иерархически.

Для проведения имитационного моделирования анализируемые БП должны быть иницированы (начаты) стартовым событием. Пользователь должен решить, каким образом и как часто необходимо производить инициацию БП, а также расставить БП по приоритетам, например, определить, какие БП должны выполняться в первую очередь. Все это описывается в *Модели инициации процесса*. Установка приоритетов осуществляется с помощью атрибута *Приоритет* в папке атрибутов *Имитация*, описывающих стартовое (начальное) событие. Значение атрибута *Приоритет* передается всем БП, которые иницируются этим стартовым событием. Модель инициации процесса является многоуровневой объектной моделью, где объект *Интервал инициации* находится на самом нижнем уровне и имеет следующие атрибуты: *Относительное время начала интервала*, *Продолжительность интервала*, *Количество экземпляров (прогонов) процесса*, *Распределение*, *Циклическая повторяемость*, *Период*. Можно разделить период времени имитационного моделирования на несколько циклов (например, рабочие дни и выходные), каждый из которых будет содержать различные интервалы. *План инициации процесса* может иметь один или более циклов и т. д.

В *Диаграмме информационных потоков* две функции могут быть взаимосвязаны с помощью объекта *поток данных*. Этот объект описывает поток данных от одной функции к другой. Для более точного описания потока данных его можно представить в виде иерархической структуры с указанием информационных объектов, которыми обмениваются функции между собой. В зависимости от степени детализации рассматриваемых функций информационные объекты могут представлять кластеры данных, типы сущностей или ERM-атрибуты.

*Диаграмма потока материалов* отображает материалы или типы материалов, которыми обмениваются функции. Работа с этой диаграммой аналогична работе с диаграммой информационных потоков. В диаграмме потока



Таблица 6.6.2














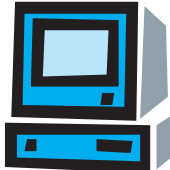
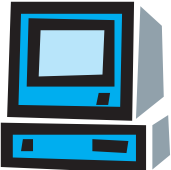
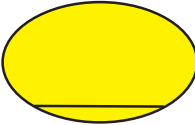
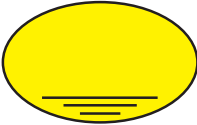


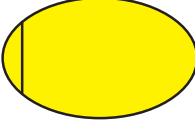



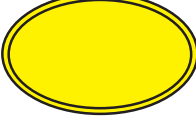








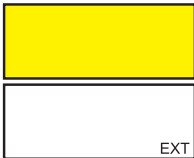


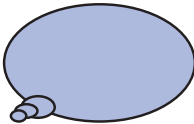


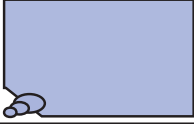


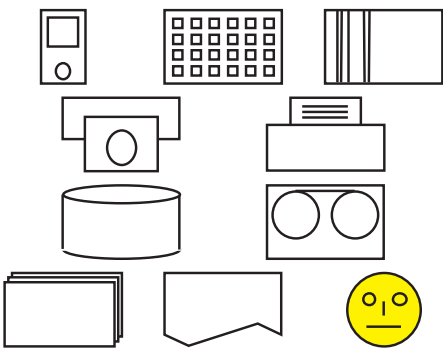
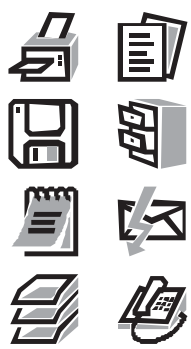

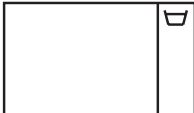











| Тип объекта             | Возможные символы типов диаграмм  |   |   |
|-------------------------|---|---|---|
|                         | eEPC  | Промышленный процесс  | Офисный процесс   |
| Событие                 |    |     |    |
| Функция                 |    |     |    |
| Правило                 |     |    |    |
| Тип прикладной системы  |    |    |   |
| Местоположение          |     |   |  |
| Организационная единица |     |   |  |
| Группа                  |    |   |  |
| Должность               |    |   |  |
| Тип сотрудника          |    |  |  |

Таблица 6.6.2 (продолжение)

| Тип объекта                                | Возможные символы типов диаграмм  |  |  |
|--|---|--|--|
|  | eEPC  | Промышленный процесс   | Офисный процесс  |
| Сотрудник                                  |    |    |   |
| Категория знаний                           |    |    |   |
| Документированные знания                   |    |    |   |
| Носитель информации                        |   |   |  |
| Тип материала                              |  |  |  |
| Тип транспортной системы                   |  |  |  |
| Тип операционного ресурса                  |  |  |  |
| Тип технического операционного обеспечения |  |  |  |
| Тип упаковочного материала                 |  |  |  |
| Складское оборудование                     |  |  |  |

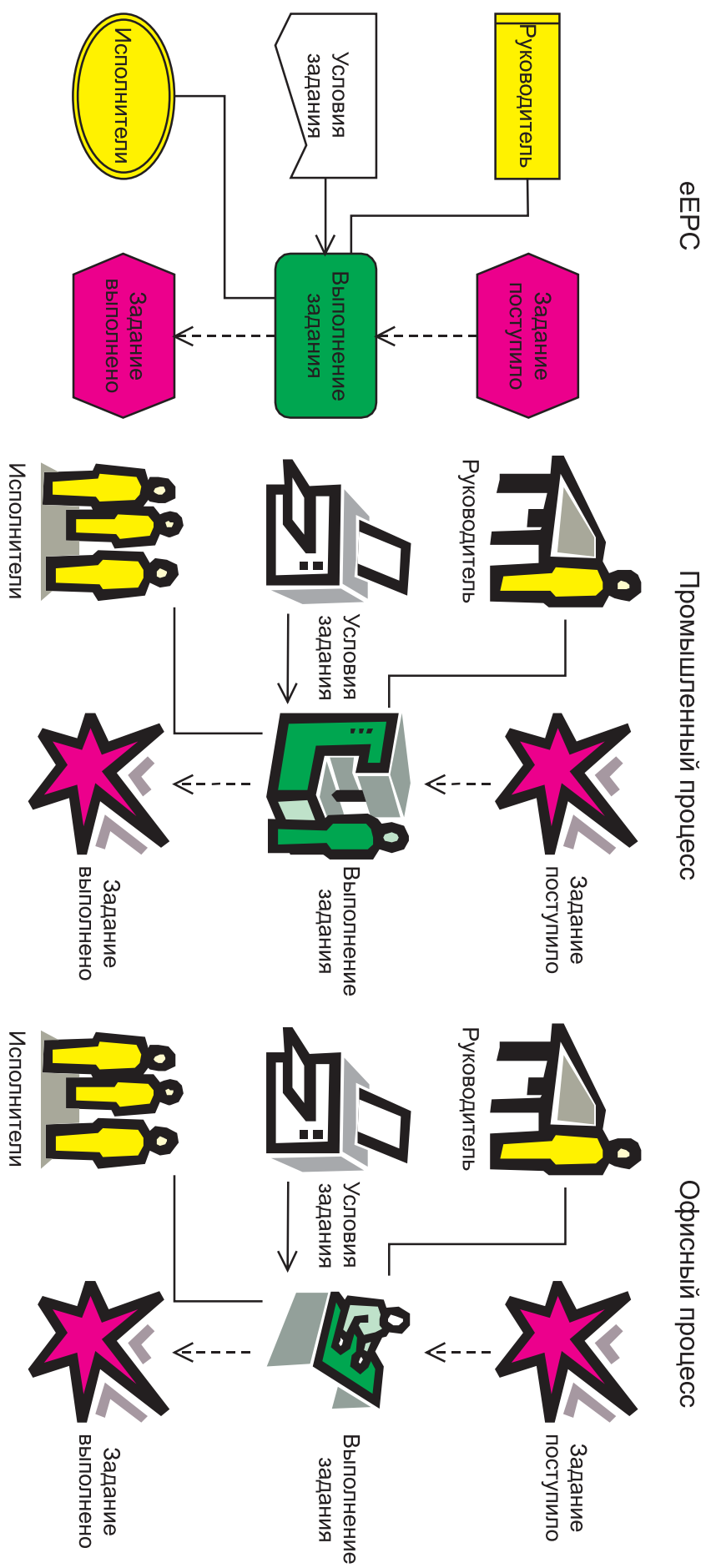


Рис. 6.6.9 Различные изображения процессных диаграмм

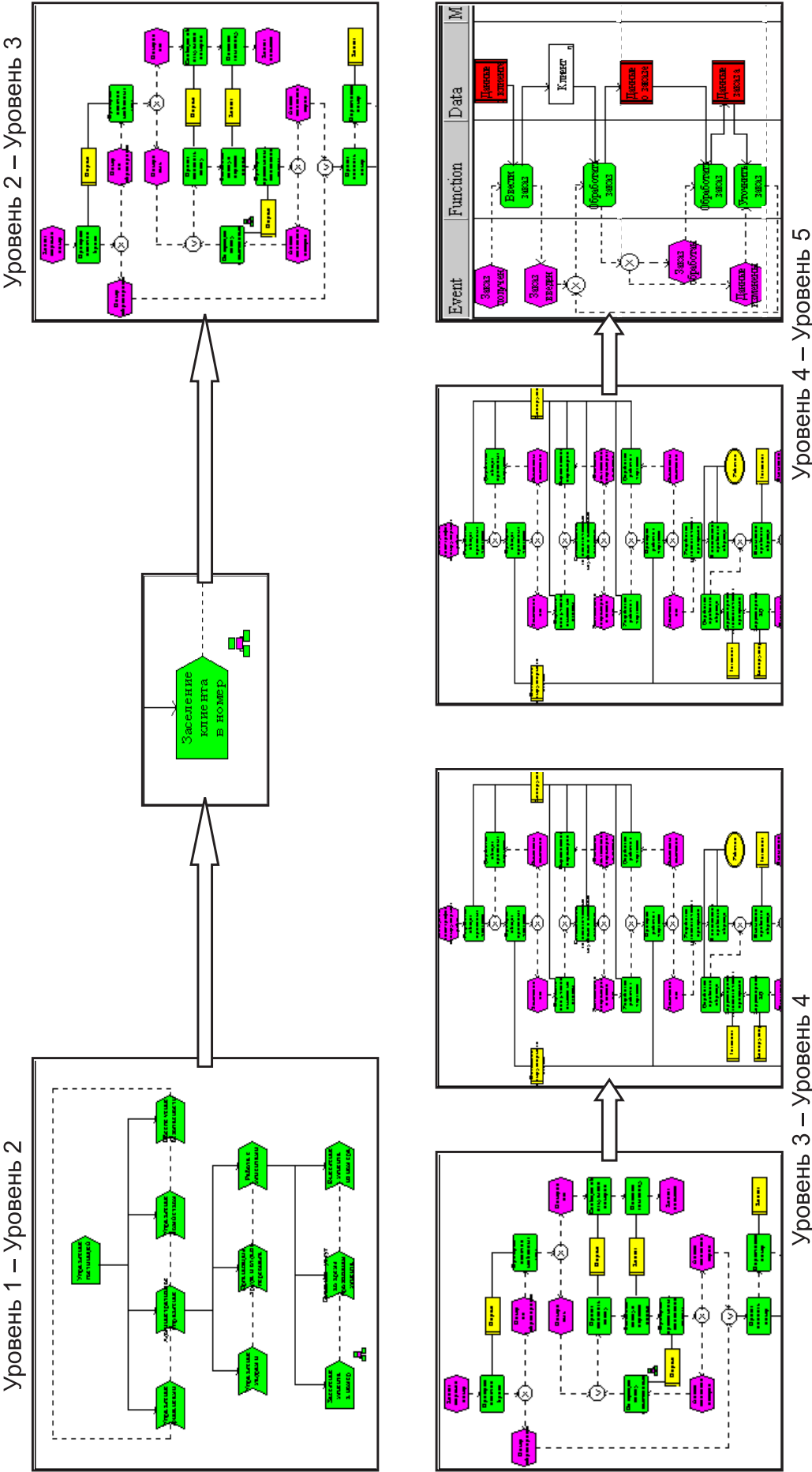


Рис. 6.6.10 Уровни детализации процессных цепочек

материалов две функции связываются посредством соединения, означающего поток материалов. Это соединение указывает поток материалов от исходной функции к целевой. Если необходимо описать поток материалов между рассматриваемыми функциями более подробно, то нужно “привязать” диаграмму потока материалов к этим соединениям, выстраивая иерархию их описаний.

Модель типа *Цепочка процесса проектирования* (PPC – Project Process Chain) связывает ARIS с MS Project и планирует ресурсы и время в рамках управления проектом, где описаны экземпляры реальных событий и конкретных функций. *Экземпляр события* – это конкретное событие, возникающее при выполнении конкретного проекта. Экземпляр события непосредственно проверяется на истинность или ложь. *Экземпляр функции* – это конкретная функция, выполняемая в рамках конкретного процесса. С экземпляром функции связывается конкретное время начала и окончания ее выполнения, а также другие необходимые атрибуты. Структурные элементы проекта (экземпляры функций, экземпляры событий, правила и соединительные линии) служат для представления хронологической последовательности в рамках проекта. Например, оператор XOR (исключающее ИЛИ) в диаграмме PPC описывает альтернативные пути выполнения проекта. Диаграмма PPC содержит такие объекты, как *Исполнитель (внутренний или внешний)*, *Операционные ресурсы*, *Общие ресурсы*. Перечисленные объекты необходимы для планирования загрузки и времени выполнения проекта. *Общий ресурс* – это ресурс, который детально не описан, и не определено, является ли он операционным или людским ресурсом. Общие ресурсы необходимы для выполнения этапов проекта.

Для описания результатов деятельности предприятий используются несколько типов моделей. С модели типа *Матрица выбора продукта* начинается описание результатов деятельности предприятия. Здесь указываются организационные подразделения и продукты, за которые эти подразделения несут ответственность. Продукты могут быть привязаны к функциям, необходимым для их производства. В модели типа *Дерево продукт/услуга* описывается сложный продукт, содержащий несколько частей, каждая из которых имеет различные составляющие части и т. д. *Диаграмма обмена продукт/услуга* предназначена для представления процесса создания продуктов (услуг), а также их обмена внутри компании. В качестве результата деятельности предприятия может рассматриваться как предоставленная услуга, так и изготовленный продукт. В диаграмме это обозначается соответствующим символом. С

помощью *Диаграммы описания продукта* проводится анализ процесса создания продукта. Эта модель может применяться для отображения того, какие ОЕ предоставляют или используют продукты, какие функции требуются для создания продукта, а также, для каких функций продукты являются входом. В диаграмме могут быть описаны цели, достижению которых способствуют различные продукты. *Дерево продукта* позволяет описать множество продуктов внешнего потребления, т. е. для реализации населению. Здесь можно описать замещающие продукты и т. д. Описанные продуктовые модели могут использоваться в самом начале моделирования, так как первым шагом проекта по оптимизации деятельности предприятия может стать выделение основных продуктов этого предприятия и выстраивание БП в соответствии с продуктовыми линиями.

На рис.6.6.11, 6.6.12 приведены примеры диаграмм типа “Матрица выбора продуктов (услуг)”.

Еще один вариант объединения данных и функций устанавливается в *Диаграмме описания функций (вход/выход)*. В этом типе диаграмм описывается преобразование входных данных в выходные и представление потока данных между отдельными функциями. Стрелки в связях определяют, используются ли информационные объекты в качестве входных данных, выходных данных или входных и выходных данных.

*Диаграмма классификации* позволяет классифицировать функции, привязывая их к классам типов объектов. Классификация может осуществляться по различным критериям. Для описания критериев классификации имеется возможность связать тип объекта *Класс типов объектов* с типом объекта *Критерий классификации*.

*Диаграмма управления бизнесом* описывает все потенциальные риски БП (функций), а также управление ими. *Управление риском* – это процесс исключения риска или уменьшения степени его негативного воздействия на БП. Диаграмма управления бизнесом по структуре аналогична матрице или таблице. Абсцисса соответствует потенциальным рискам БП, а ордината показывает возможные способы управления рисками. Решения по рискам в качестве операторов размещаются между риском и его управлением. В эту модель дополнительно могут быть включены ОЕ (с учетом требований пользователей) и документы, которые поддерживают реализацию управления соответствующим риском. Этот тип диаграммы используется, например, для описания стандартных процессов SAP.

*Структурная модель* предназначена, в основном, для обобщения, детализации или другого системного упорядочения фактов, например, иерархического. Структур-

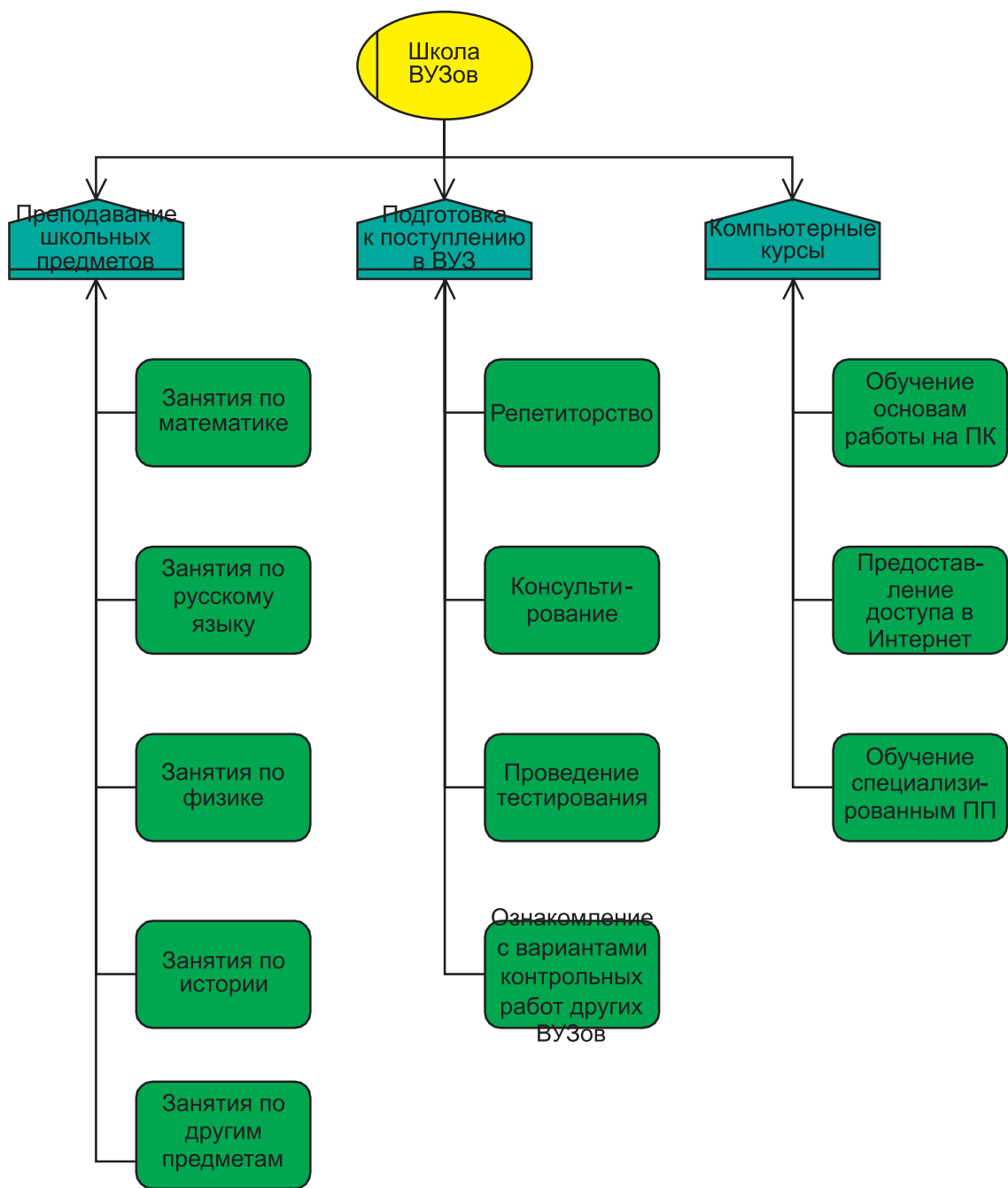


Рис. 6.6.11 Матрица выбора услуг, предоставляемых Школой ВУЗов при МФ ЮУрГУ



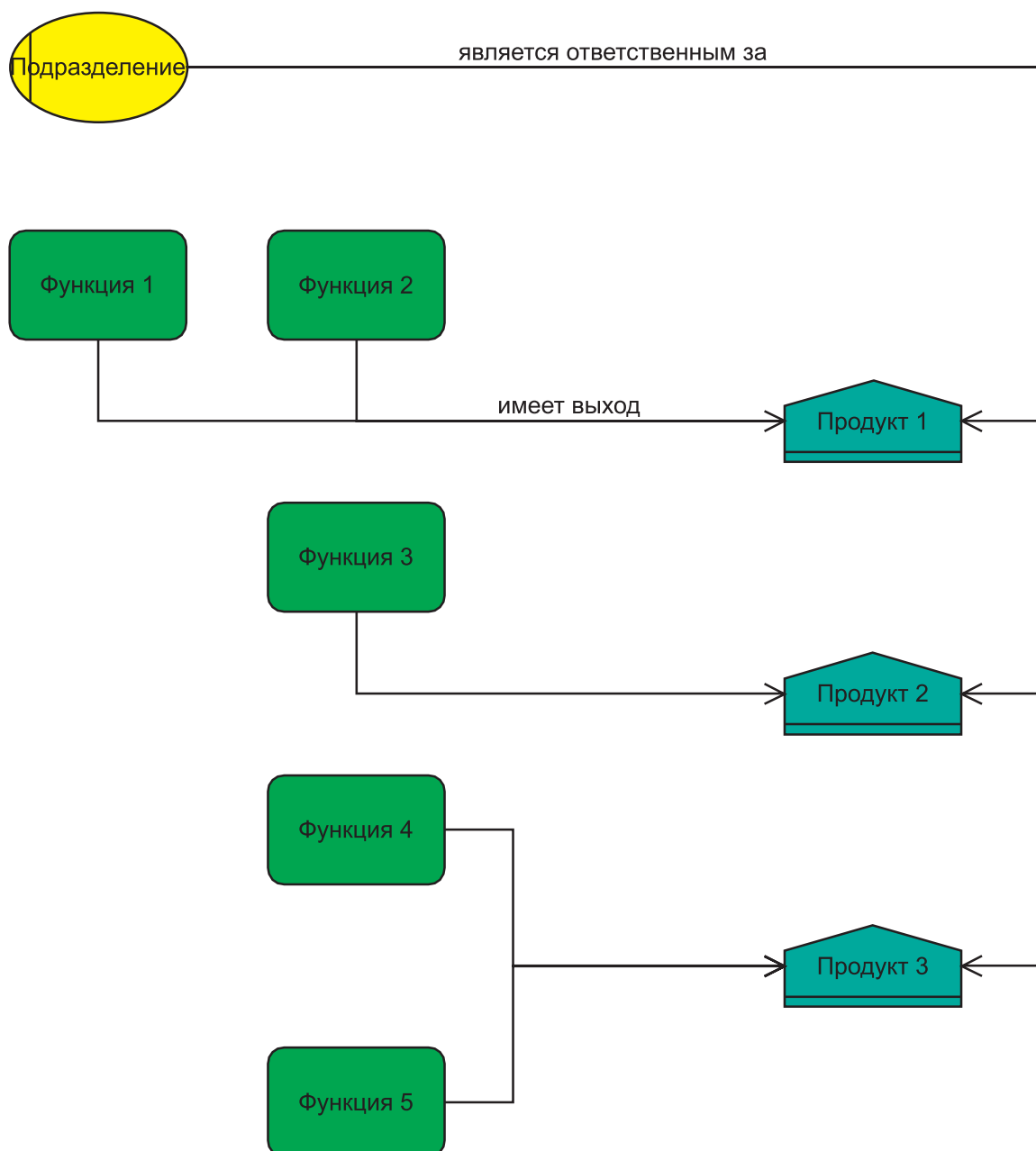


Рис. 6.6.12 Диаграмма матрицы выбора продукции

ный элемент представляет некоторый факт (в соответствии с предполагаемой систематизацией). Структурные модели применяются в управлении качеством, особенно для целей сертификации, где они показывают норму каждой отдельной ее компоненты, а модели, которые наиболее полно удовлетворяют критериям качества, “привязываются” к отдельным структурным элементам. ARIS-отчеты по этой модели позволяют легко анализировать и оценивать эти факты, а также использовать их при документировании процессов управления качеством.

### Этапы спецификации и реализации системы и ее БП

На этапе спецификации типы моделей процессного вида объединяют те модели или их объекты, которые рассматривались при спецификации в других видах моделирования (организационном, функциональном и информационном). В моделях, которые объединяют функции и данные описываются потоки данных между прикладными системами (ПС) или типами модулей (ИТ-функциями). Для этого поток данных в модели размещается между отдельными ПС или типами модулей.

В моделях, которые объединяют ОЕ и данные, решаются две основные задачи. Во-первых, определяют ОЕ, которые ответственны за конкретные объекты данных. Во-вторых, указывается, каким ОЕ разрешен доступ к отдельным объектам данных. Для авторизации доступа с объектами данных связываются либо должности, либо типы людских ресурсов. При помощи привязки типов людских ресурсов может быть указано, что к конкретному полю объекта данных могут иметь доступ только конкретные сотрудники бухгалтерии. Кроме авторизации доступа, определяется ответственность за содержимое поля. Для этого в *Диаграмме отношений* может быть введена связь между ОЕ и объектом данных, называемая *ответственный за*.

В моделях, которые объединяют ОЕ и функции, описываются факты, дающие исчерпывающие ответы на следующие вопросы: “Кто (ОЕ, должность, людские ресурсы и т. д.) несет ответственность за типы ПС и типы модулей, которые были определены в функциональной модели на уровне спецификации, и кто использует эти ПС? На каких рабочих местах компании (организационная модель) какие типы ПС и типы модулей должны использоваться? Какие платформы, доступные в компании (типы компонент АО (организационная модель)), для каких типов ПС являются подходящими?” Например, для того, чтобы ответить на первый вопрос, в *Диаграмме доступа* достаточно установить связь между элементами организационной схемы (ОЕ,

должности и людские ресурсы) и объектами диаграммы ПС (тип ПС, тип модуля, ИТ-функции и т. д.). Причем, один тип связи может описывать то, что ОЕ ответственна за разработку типа ПС, другая связь указывает на то, что ОЕ использует тип ПС и т. д. Вопрос местонахождения (рабочего места) можно описать посредством установки связей объектов организационной модели с типами ПС, типами модулей и типами ИТ-функций. При спецификации имеют дело не с индивидуальными ПС (в смысле индивидуальных лицензий на них), а с типами ПС. Поэтому эти связи не определяют местоположение реальных ПС (такая привязка осуществляется на этапе реализации), но указывают возможные местоположения ПС конкретного типа.

Типы компонент АО, доступные в компании, определяются на уровне спецификации в организационной модели. В процессном виде моделирования устанавливаются отношения между этими типами компонент АО и типами ПС. Таким образом, описываются аппаратные платформы, на которых могут работать определенные типы ПС, типы модулей или типы ИТ-функций. На этом этапе могут быть также описаны типы персональных компьютеров, операционных систем и СУБД, включаемые в функциональную модель.

Определенные типы диаграмм (например, *Структурные схемы программ – PSP*) используются для проектирования (сборки из компонент) программных систем. Они позволяют просматривать все аспекты проектирования системы в их взаимосвязи. Диаграмма типа *Структурная схема программы* позволяет моделировать все взаимосвязи типов ПС, типов модулей и типов ИТ-функций при помощи других типов ARIS-диаграмм. Здесь имеется возможность отображать логическую последовательность событий для управления перечисленными типами объектов. Аналогично связи функций и событий в ЕРС можно определить последовательность модулей и в *Структурной схеме программы*, где событие рассматривается как триггер, переключающий типы модулей или типы ПС. Для отображения ветвлений можно использовать логические операторы и т. д.

На этапе спецификации используется модель типа *Блок-схема программы*, в которой описывается последовательность процедур в программе. Порядок процедур определяется связями между объектами. На рис.6.6.13 приведены объекты и их связи, которые используются в диаграммах типа *Блок-схема программы*. На рис.6.6.14 приведен простейший пример процедурной последовательности. Блок-схема ориентирована на реализацию и дает ясное представление о том, как выполняется программа и что в ней делается.

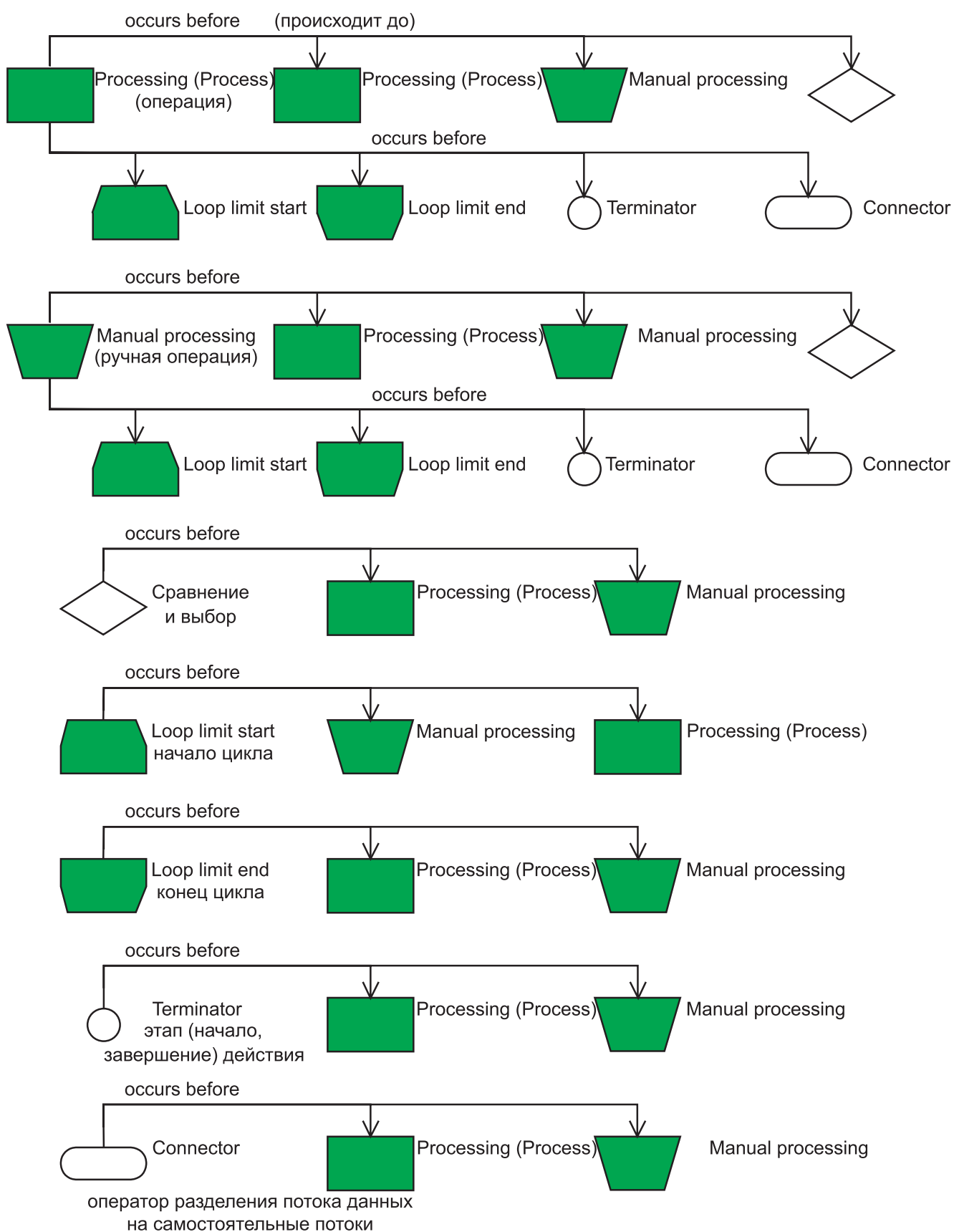


Рис. 6.6.13 Символы типов объектов и связей в моделях типа Program flow-chart (PF)

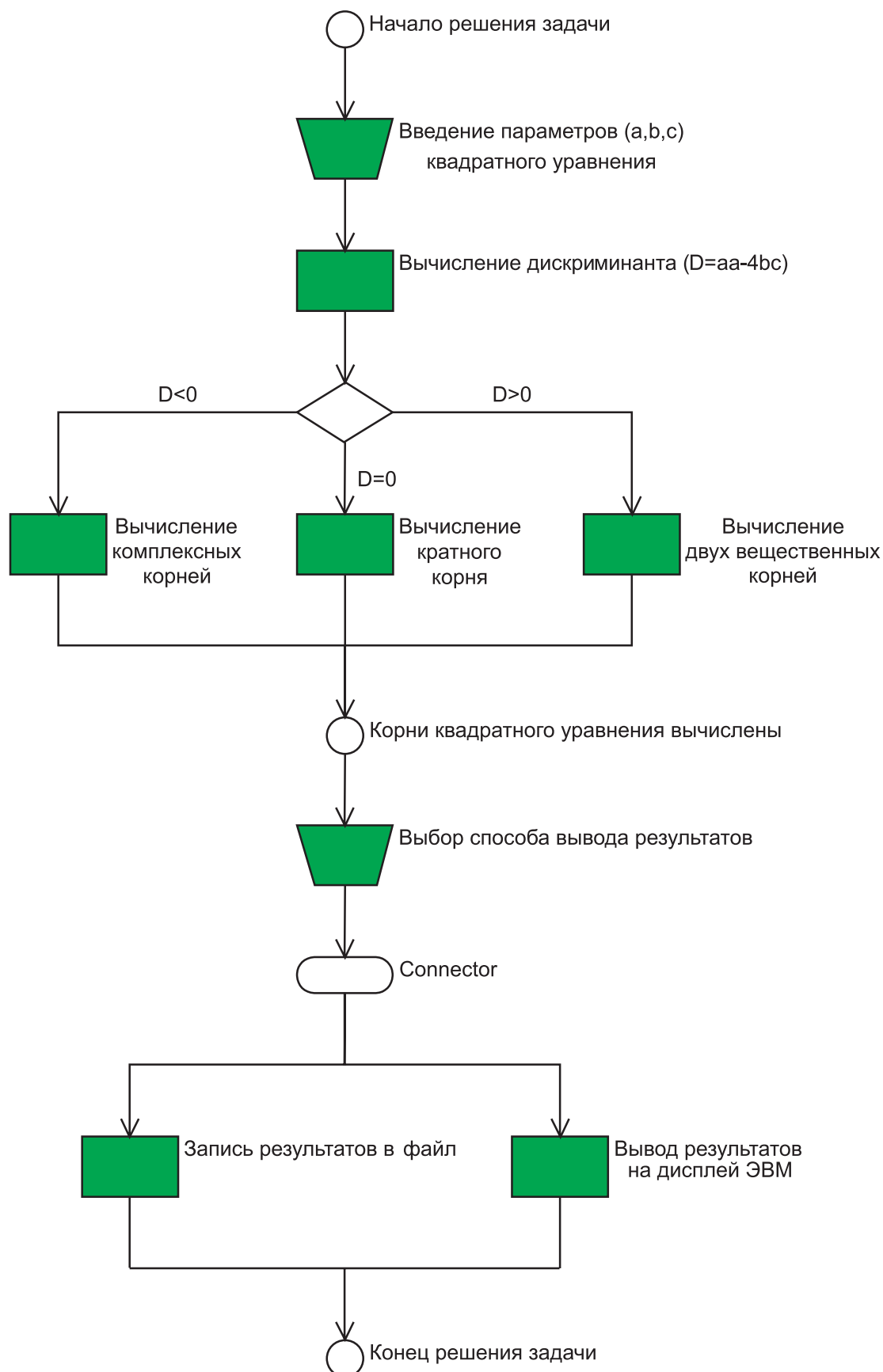


Рис. 6.6.14 Блок-схема программы вычисления корней квадратного уравнения

*Диаграмма экрана* на этапе спецификации используется для описания экранных форм при разработке программного обеспечения. Диаграммы экранов описывают структуру и в некоторой степени функциональность экранной формы. Эта модель используется, например, для автоматизации процесса генерации экранных форм из диаграмм экранов.

Вопросы, рассмотренные при спецификации в рамках процессного вида моделирования, имеют также отношение к этапу описания реализации. В отличие от этапа спецификации здесь рассматриваются не типы объектов, а сами объекты и конкретные лицензии ПС. Компоненты АО на этапе реализации однозначно идентифицируются посредством инвентарных номеров, принятых в компании. Заметим, что объект *Компонента* АО связывается с различными представлениями информационных объектов в зависимости от этапа описания. Для этапа реализации это таблицы, поля и т. д. Для этапа спецификации это отношения, атрибуты и т. д. Для формулировки требований это типы сущностей, кластеры данных и т. д.

При спецификации связь между типом ПС и местоположением позволяет определить, какие типы ПС могут быть расположены на конкретных рабочих местах в компании. Для того, чтобы фактически описать, в каких местах компании используются индивидуальные лицензии на приобретенные типы ПС, можно на этапе реализации связать местоположение с типами объектов *Прикладная система, Модуль и ИТ-функция в Диаграмме доступа (физического)*.

## Поддерживаемые методы моделирования и ARIS-интеграция

ARIS поддерживает объектно-ориентированное моделирование, в котором используются графические обозначения *Техники объектного моделирования* (OMT), разработанной для представления различных точек зрения при описании систем. Для этой цели применяются следующие методы: объектное моделирование, динамическое моделирование, функциональное моделирование. *Объектное моделирование* – представление статичных, структурных, а также ориентированных на данные аспектов системы, позволяющие отображать структуру объектов, их отношения к другим объектам и их атрибуты. *Динамическое моделирование* – отображение временных, поведенческих и управляющих аспектов системы, позволяющее описывать последовательность операций с помощью последовательности событий. *Функциональное моделирование* – отображение временных и функциональных аспектов системы, с помощью которого можно описывать преобразование зна-

чений. Каждая из этих моделей содержит перекрестные ссылки на другие модели. Например, объектная модель описывает структуры данных, которые используются в динамической и функциональной моделях. Процессы в функциональной модели соответствуют операциям в объектной модели и т. д.

Следует заметить, что OMT, наряду с другими методами, интегрирован и преобразован в унифицированный язык моделирования – Unified Modeling Language (UML). В ARIS реализованы основные типы UML-моделей, например, UML-диаграмма активности, UML-диаграмма класса, UML-диаграмма описания класса, UML-диаграмма сценариев, UML-диаграмма компонент, UML-диаграмма состояний, UML-диаграмма взаимодействия. Кратко опишем некоторые из них.

Тип модели *UML-диаграмма класса* отражает статичные отношения между классами, объектами и интерфейсами. Например, к классу с помощью отношения *иметь участником* могут быть привязаны соответствующие операции (методы) и атрибуты (смотрите рис.6.6.15, 6.6.16). *UML-диаграмма активности* описывает процесс как последовательность действий над объектами. Эти диаграммы действий привязываются к классу, операции или приложению и описывают относящийся к ним внутренний процесс. *UML-диаграмма состояний* подобно UML-диаграмме активности описывает аналогичные аспекты, но она сфокусирована на состояниях объектов и описывает действия, относящиеся к состоянию. *UML-диаграмма взаимодействия* описывает взаимодействия в форме обмена сообщениями между объектами, где объекты (экземпляры) – это конкретные экземпляры классов. *UML-диаграмма компонент* описывает аспекты реализации, например, структуру кода (компонент), структуру работы системы и т. д.

Различные ARIS-модели связаны друг с другом. Для этого используются два основных механизма. Во-первых, одни и те же объекты могут использоваться в различных моделях. Например, тип объекта *Функция* существует как в *eEPC*, так и в модели типа *Функциональное дерево*. Одни и те же типы объектов могут изображаться различными символами и именами символов в различных моделях. Это происходит обычно в том случае, когда символ и его имя используются в одном типе, но объекты с той же семантикой существуют в моделях других типов. Например, *Функции* (*eEPC*, *Функциональное дерево* и т. д.) и *Операции* (в UML-диаграмме класса) имеют различные символы (графические изображения), но оба понятия отображают один и тот же тип объекта (их атрибуты идентичны). Идентичные объекты могут появиться в результате операций *Copy* (Копировать) и *Paste* (Вставить) или при

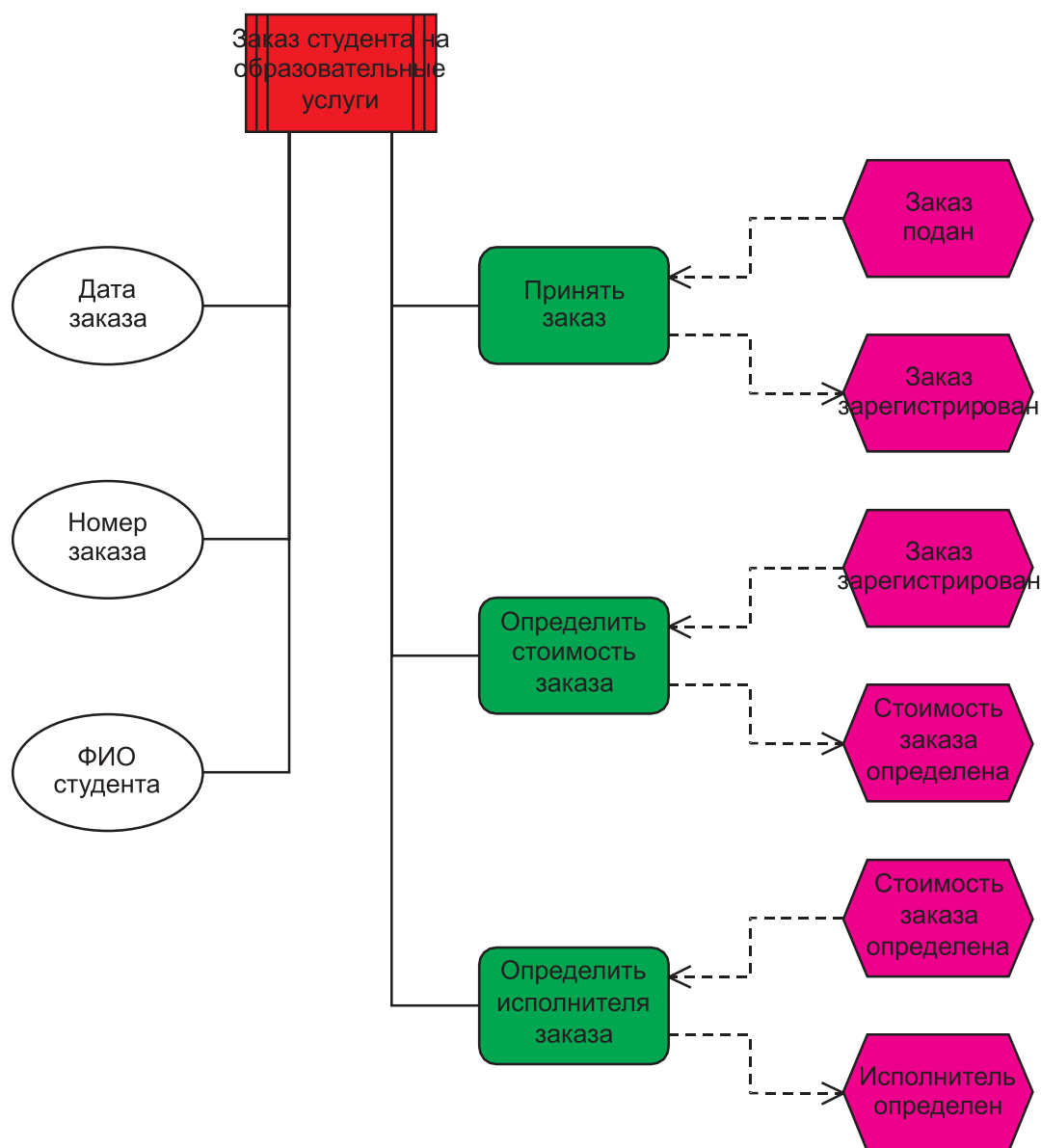


Рис. 6.6.15 Диаграмма класса “Заказ студента на образовательные услуги”



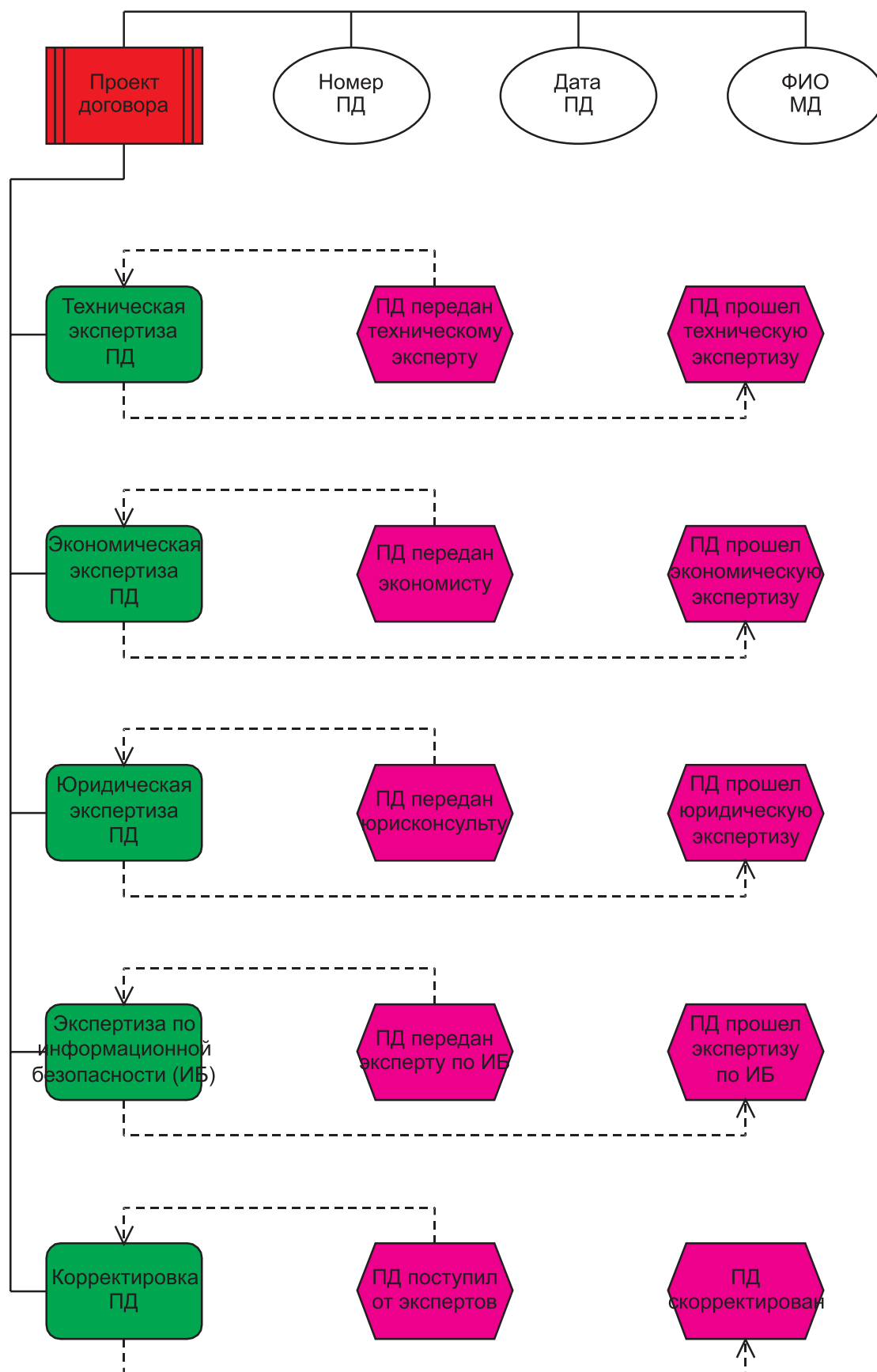


Рис. 6.6.16 Диаграмма класса “Проект договора (ПД)”

создании объекта с тем же именем, что и существующий объект, в диалоговом окне Object Selection (Выбор Объекта). Многие типы объектов в UML-диаграммах идентичны типам объектов в других ARIS-диаграммах. Во-вторых, связь между двумя моделями осуществляется путем привязки объекта одной модели к другой модели. Для этого используется команда Create Object Assignment (Создать привязку объекта) в диалоговом окне Properties (Свойства) Object (Объекта) вкладки Assignment (Привязка). С помощью привязки модели уточняется описание того объекта, к которому модель привязывается.

Интеграция UML-диаграмм с другими ARIS-моделями достаточно глубокая. Например, диаграмма eEPC содержит множество типов объектов, которые также используются в UML-диаграммах. Причем, некоторые типы объектов eEPC идентичны типам объектов UML, и единственным различием между ними является графическое представление. Классы с помощью связи *является входом для* могут быть использованы в eEPC как источники информации для функций. Если требуется более грубое или более детальное описание входа и выхода функций, то достаточно связать функции с пакетами (для грубого описания) или атрибутами (для детального описания). Вместо отображения физического чтения или записи в этом случае будет представлено, каким образом создается или используется информация. Функция может быть реализована с помощью одного или нескольких классов операций. Для этого нужно либо соединить функции и операции с помощью связи *вызывает*, либо вследствие идентичности типа объекта использовать функции непосредственно как операции в UML-диаграмме класса. Диаграмма eEPC позволяет описывать процесс в пределах одного класса или одной операции. Для этих целей eEPC может быть привязана к обоим типам объекта и т. д.

## 6.7. Моделирование знаний

В ARIS 5.0 добавлены новые типы объектов – *Категория знаний (Knowledge Category)* и *Документированные знания (Document Knowledge)*, а также новые типы моделей – *Диаграмма структуры знаний (Knowledge Structure Diagram)* в моделировании данных и *Карта знаний (Knowledge Map)* в процессном виде моделирования. Кроме того, существующие типы моделей, описывающих БП (eEPC, PCD, Офисный БП и т. д.), расширены включением конструкций для создания, обработки, использования и управления знаниями. Новые типы объектов и моделей интегрируются с наиболее важными типами моделей уровня определения требований (такими как eEPC,

eERM, организационная диаграмма, дерево функций). Это позволяет при оптимизации БП использовать, анализировать и улучшать модели управления знаниями.

Тип объекта *Категория знаний* графически изображается овалом с клапаном и содержит конкретные знания. Примерами *Категорий знаний* являются знания об управлении проектом, конкретные знания о производстве и его технологиях, знания о клиенте и конкурентах и т. д. Эти категории помогают в классификации знаний, которыми компания обладает или которые ей необходимы.

Знания, распределяемые по отдельным категориям, могут быть, во-первых, не явными (их невозможно полностью документировать), во-вторых, знаниями служащего или группы в форме способности и опыта, в-третьих, явными знаниями (их можно документировать), например, техническая документация, чертежи и т. д. Категории знаний часто содержат более одного вида знаний. Например, знания об управлении проектом могут включать опыт руководства проектом и советы по управлению проектом.

В дополнение к общим атрибутам (Имя, Описание, Замечание, Создатель и т. д.) в ARIS 5.0 используются атрибуты, которые служат для более подробного описания *категорий знаний*. Перечислим и кратко охарактеризуем эти атрибуты.

Атрибут *частота обновления* имеет *перечисляемый тип* (ежечасно, ежедневно, еженедельно, ежемесячно, ежегодно, редко, никогда). Частота обновления описывает, как часто знание должно быть обновлено, чтобы быть актуальным. Основные математические знания, например, обновляются редко, а знание тенденций изменения биржевого курса должно обновляться ежедневно или даже ежечасно.

Атрибут *значимость* имеет *тип процент* и меняется от 0% (полностью незначимые) до 100% (чрезвычайно важные).

Атрибут *степень охвата* имеет *тип процент* и для значимого знания может меняться от 0% (не охваченное вообще) до 100% (максимально возможная степень охвата). Для изображения на *карте знаний* степени охвата категории знаний организационной единицей или личностью используется соответствующий атрибут связи типа “распоряжается”.

Атрибут *преимущество знания* имеет *тип процент*. Относительное положение компании среди конкурентов в терминах знаний может меняться от 0% (конкуренты имеют максимально возможное преимущество перед вашей компанией) до 100% (ваша компания имеет максимально возможное преимущество перед конкурентами).

Атрибут *использование знаний* имеет *тип процент*. Степень использования конкретной категории

знаний может меняться от 0% (значимые знания не используются вообще) до 100% (максимальное использование значимых знаний).

Атрибут *желаемая степень охвата* имеет тип *процент* и для значимых знаний может меняться от 0% (неохваченные вообще) до 100% (максимально возможная степень охвата).

Перечисленные атрибуты описывают значимые *категории знаний* компании и могут использоваться при совершенствовании управления знаниями.

Тип объекта *документированные знания* включает в себя те *категории знаний*, которые явно задокументированы, или их документирование принципиально возможно. Например, знания об использовании программного обеспечения в компании задокументировать можно. А вот знания психологических особенностей субъектов ответственности предприятия (личностей) или клиентов фирмы вряд ли можно задокументировать. Многие факты нельзя документировать из этических соображений или из-за правовых ограничений.

Анализ общих и документированных знаний помогает установить возможности и ограничения информационной поддержки деятельности компании. При этом большую пользу оказывают документированные знания, т. к. они могут быть сохранены, переданы, обработаны и многократно использованы.

Тип объекта *документированные знания* изображается прямоугольником с клапаном в левом нижнем углу. Он содержит те же самые типы атрибутов, как и тип объекта *категория знаний*.

С помощью *диаграммы структуры знаний* можно размещать категории знаний по их содержанию в подгруппы. Категория знаний может включать другие категории знаний, а также документированные знания. Последние могут быть разделены на несколько подкатегорий документированных знаний, которые не могут включать общие (недокументированные) знания. Для документированных знаний в диаграмме структуры знаний можно показать информационные средства, с помощью которых знания задокументированы, или указать, какие информационные объекты модели данных (или классы объектно-ориентированной системы) служат для документирования этих знаний. Аналогично моделируются типы или классы прикладных систем, которые используются для управления знаниями.

*Карта знаний* изображает распределение различных категорий знаний. Различные типы объектов в организационном представлении (например, Организационная единица, Личность, Группа и др.) могут быть соединены с категориями знаний при помощи связи

типа *располагается*. В дополнении к тому, что конкретная личность или организационная единица обладает знаниями в конкретной категории, можно показать степень охвата этих знаний. Связь *располагается* содержит атрибут *степень охвата*, который может выражать в процентах степень охвата знаний выбранной категории для рассматриваемой организационной единицы. Значение 100% показывает максимальный охват, а значение 0% описывает абсолютное отсутствие знаний выбранной категории у рассматриваемой организационной единицы. В дополнение к этой количественной мере можно рассмотреть качественную оценку в форме графика. Для этого имеется атрибут *качество охвата*, который может принимать следующие значения: *низкое, среднее, высокое* и *максимальное*. Эта информация может представляться с использованием графических символов в связях.

Значение атрибутов *степень охвата* и *качество охвата* не взаимосвязаны. Если используются оба атрибута, то предполагается, что качеству охвата в значении *низкое* соответствует степень охвата менее 25%, значению *среднее* – степень охвата 25 – 50%, значению *высокое* – степень охвата 50 – 75%, и значению *максимальное* – степень охвата 75 – 100%. *Карта знаний* ориентирована на организационные единицы, т. е. каждой организационной единице поставлены в соответствие значимые *категории знаний*. Но можно поставить *категории знаний* в центр *карты знаний* и моделировать значимые организационные единицы вокруг них.

Создание и использование знаний в БП компании моделируется с помощью типов моделей для представления БП (еЕРС, еЕРС с материальным потоком, Офисный БП, Индустриальный БП, PCD, PCD с материальным потоком). Типы объектов *категории знаний* и *документированные знания* доступны в этих диаграммах. Можно определить, какого вида знания (общие или документированные) являются необходимыми для выполнения функций, и отметить, какое знание создано (задокументировано) при выполнении функции. Этот тип представления включает возможность изучения БП в терминах обработки знаний. Например, могут быть обнаружены пробелы в необходимых знаниях. Кроме того, может быть определена квалификация субъекта ответственности, необходимая для выполнения конкретной функции.

Описание БП, в которые включены соответствующие знания, документы и инструкции, можно использовать для введения в курс дела новых служащих или временно заменяющих работников.

Знания являются важнейшими управляемыми ре-

сурсами, такими же, как и классические экономические ресурсы. Управление знаниями – это сумма всех методов и мероприятий (сбор, представление, распространение и т. д.), используемых организацией для того, чтобы создавать знания и делать их доступными в любое время и на любом рабочем месте для тех исполнителей, которые используют эти знания в своей профессиональной деятельности. Цель управления знаниями состоит в их накоплении и оптимальном использовании.

Рассмотрим основные задачи управления знаниями и ARIS-средства их поддержки.

**Задача: Классификация знаний.** Цель – классифицировать интеллектуальный капитал организации, т. е. описать различные области и группы знаний, например, для проектирования структуры хранения знаний. **Поддержка: Диаграмма структуры знаний** показывает, как база знаний организации разделена на различные категории знаний, какие категории знаний документированы и т. д. Для документированных знаний можно также отобразить их носителей информации, т. е. показать, где эти знания сохранены.

**Задача: Организационное распределение знаний.** Цель – обеспечить доступность знаний в компании. **Поддержка:** тип диаграммы *Карта знаний* используется для отображения организационного распределения различных категорий знаний. Она может показать, какая организационная единица, офис или служащий обладает “ноу-хау” в некоторых категориях знаний и каков уровень их компетенции.

**Задача: Обработка знаний в БП.** Цель – эффективно использовать знания. **Поддержка:** диаграмма eEPC, офисный и индустриальный БП, диаграмма распределения функций, т. е. все типы моделей, которые содержат категории знаний и документированные знания.

На рис.6.7.1 – 6.7.3 приведены примеры диаграмм, которые наглядно демонстрируют особенности использования знаний и работы с ними. Карта знаний, приведенная на рис.6.7.1, имеет структуру, ориентированную на организационные единицы. Для карт знаний часто используется матричное представление. На рис.6.7.2. приведен соответствующий пример. На этом рисунке также показано альтернативное визуальное представление для различных степеней охвата, где менее охваченные категории знаний изображаются графическими фигурами меньшего размера.

На рис.6.7.4 приведен пример использования и обработки знаний в процессных моделях. На рис.6.7.5, 6.7.6 приведены типы объектов и их связей, используемых при построении моделей управления знаниями.

## 6.8. Использование моделей

Перечислим основные задачи и подзадачи, а также типы диаграмм (моделей), которые предоставляет ARIS для их решения.

**Задача – документирование деятельности компании.** **Подзадачи** – документирование целей компании, документирование организационной структуры, документирование БП. **Типы диаграмм** – диаграмма целей, Y-диаграмма, дерево функций, диаграмма цепочки добавленного качества, офисный БП, производственный БП, eEPC, PCD, организационная диаграмма.

**Задача – администрирование БД и хранение данных.** **Подзадачи** – структурирование данных, проектирование БД, администрирование БД, управление доступом. **Типы диаграмм** – eERM, SAP SERM model, SeDaM model, UML-модели данных, диаграмма отношений, диаграмма таблиц, диаграмма описания классов.

**Задача – управление сетями и средствами аппаратного (компьютерного) обеспечения.** **Подзадачи** – формулирование требований к информационным технологиям (ИТ), документирование ИТ, распределение прав доступа к ИТ. **Типы диаграмм** – топология сети и диаграмма сети.

**Задача – функционально-стоимостный анализ БП.** **Подзадачи** – описание БП и организационных структур как исполнителей функций, анализ центров затрат, вычисление стоимости БП. **Типы диаграмм** – eEPC, PCD, организационная диаграмма, CD-диаграмма, диаграмма категорий затрат.

**Задача – управление качеством.** **Подзадачи** – описание структуры TQM, описание процессов сертификации, подготовка документации по сертификации. **Типы диаграмм** – дерево продукции, матрица выбора продукции, eEPC, PCD, диаграмма офисного БП, диаграмма производственного БП, диаграмма цепочки добавленного качества, структурные модели, организационная диаграмма.

**Задача – реорганизация деятельности предприятия.** **Подзадачи** – подготовка проектной документации, реализация реорганизации. **Типы диаграмм** – диаграмма целей, диаграмма цепочки добавленного качества, eEPC, PCD, организационная диаграмма, модели продукции, модель выпускаемой продукции.

**Задача – внедрение SAP R/3.** **Подзадачи** – подготовка проекта внедрения SAP R/3, стадия анализа и спецификации, стадия проектирования (бизнес-планирование). **Типы диаграмм** – eEPC, организационная диаграмма и т. д.

**Задача – проектирование и реализация программного обеспечения.** **Подзадачи** – подготовка проектной документации, определение прикладных систем и модулей, описание алгоритмов вычислений, проектирование

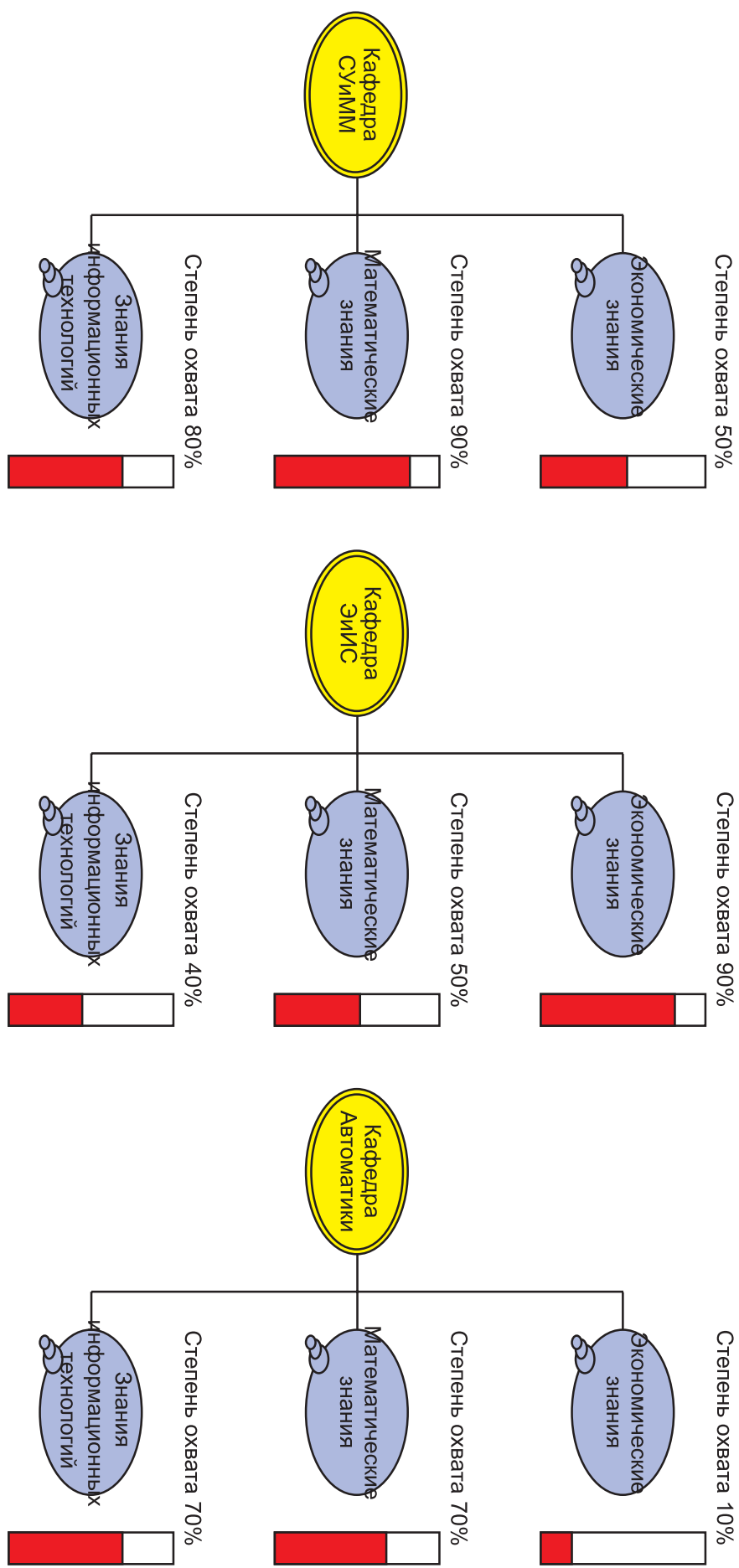


Рис. 6.7.1 Диаграмма типа "Карта знаний"

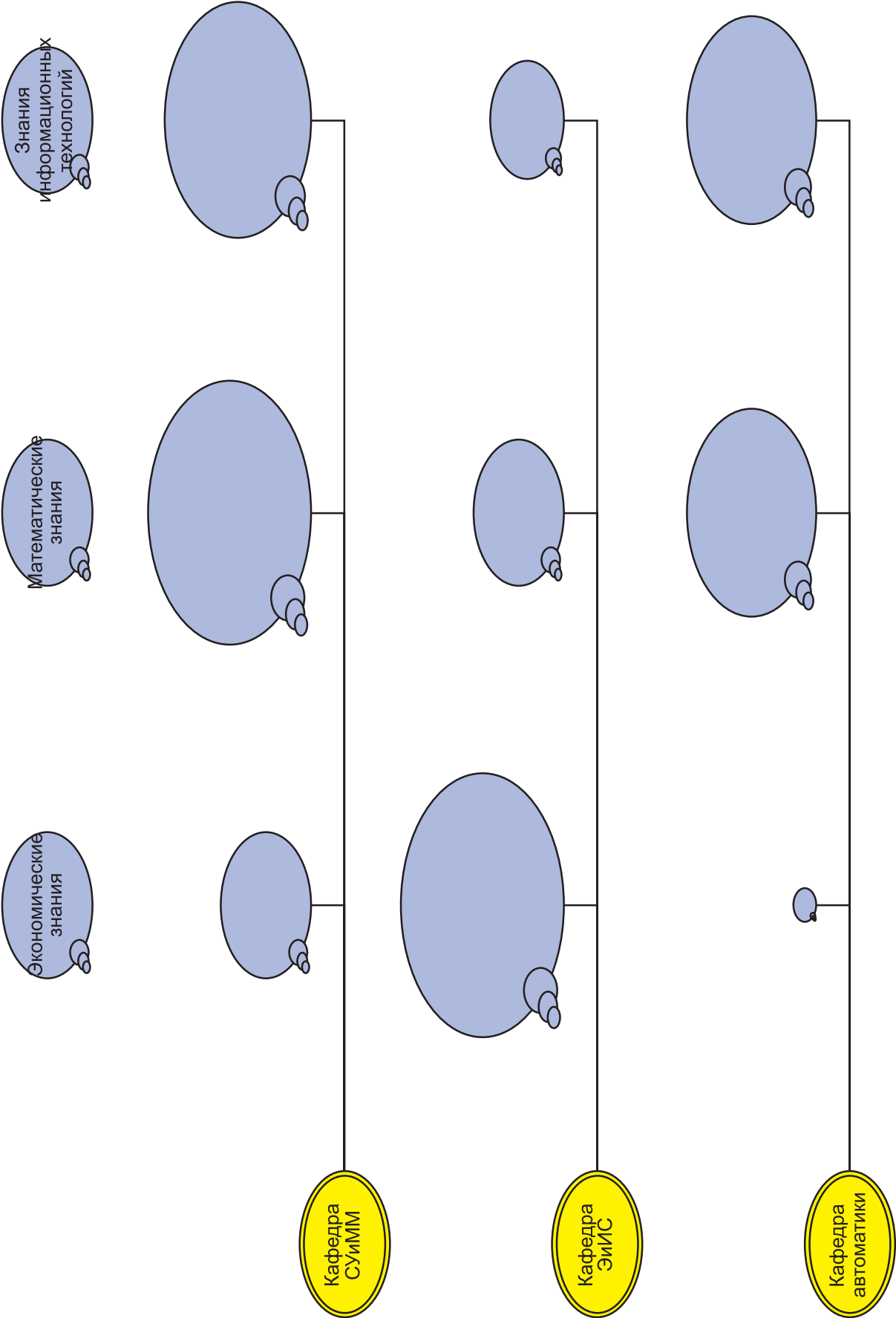


Рис. 6.7.2 Матричное представление карты знаний



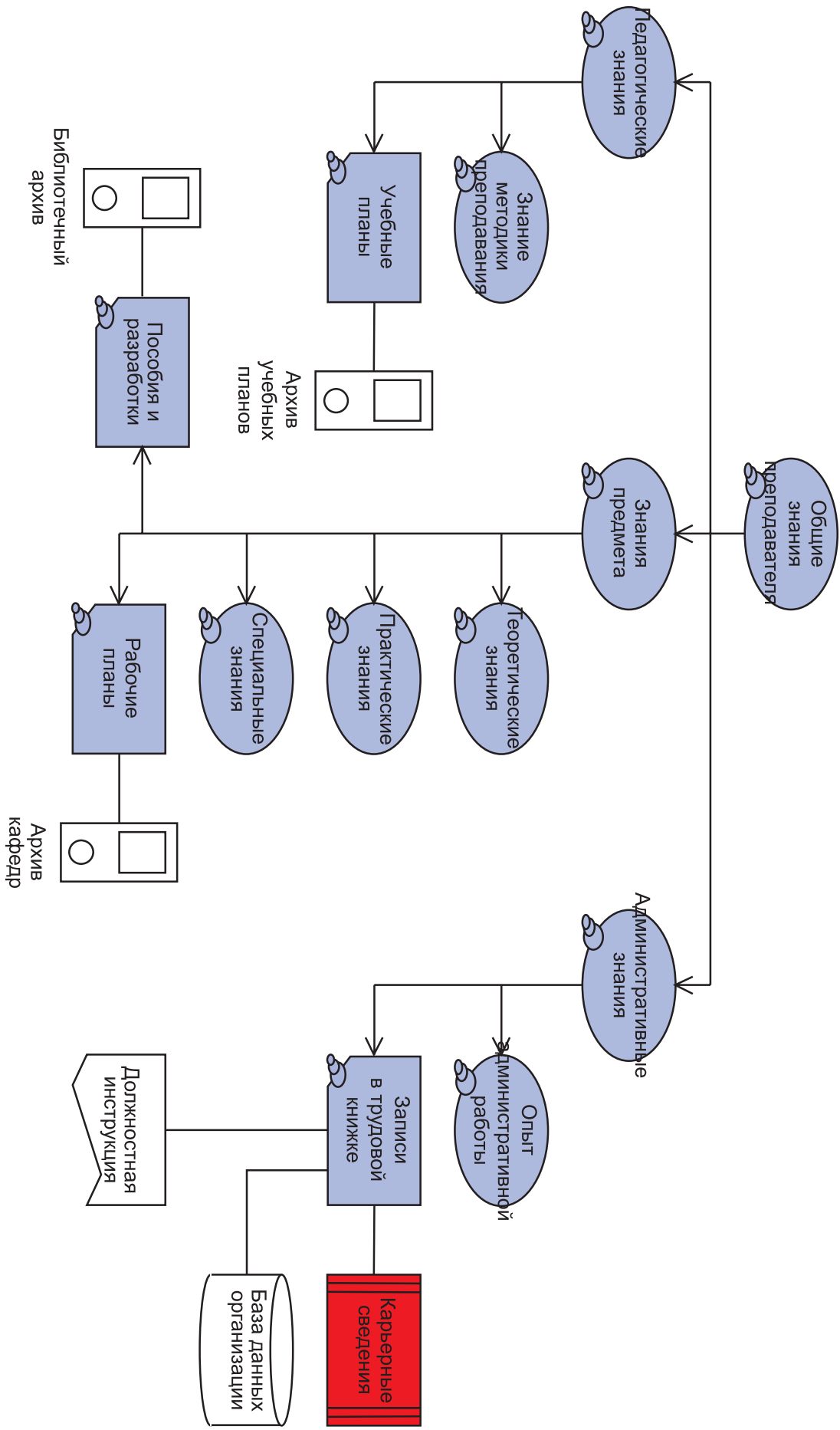


Рис. 6.7.3 Структурная диаграмма знаний

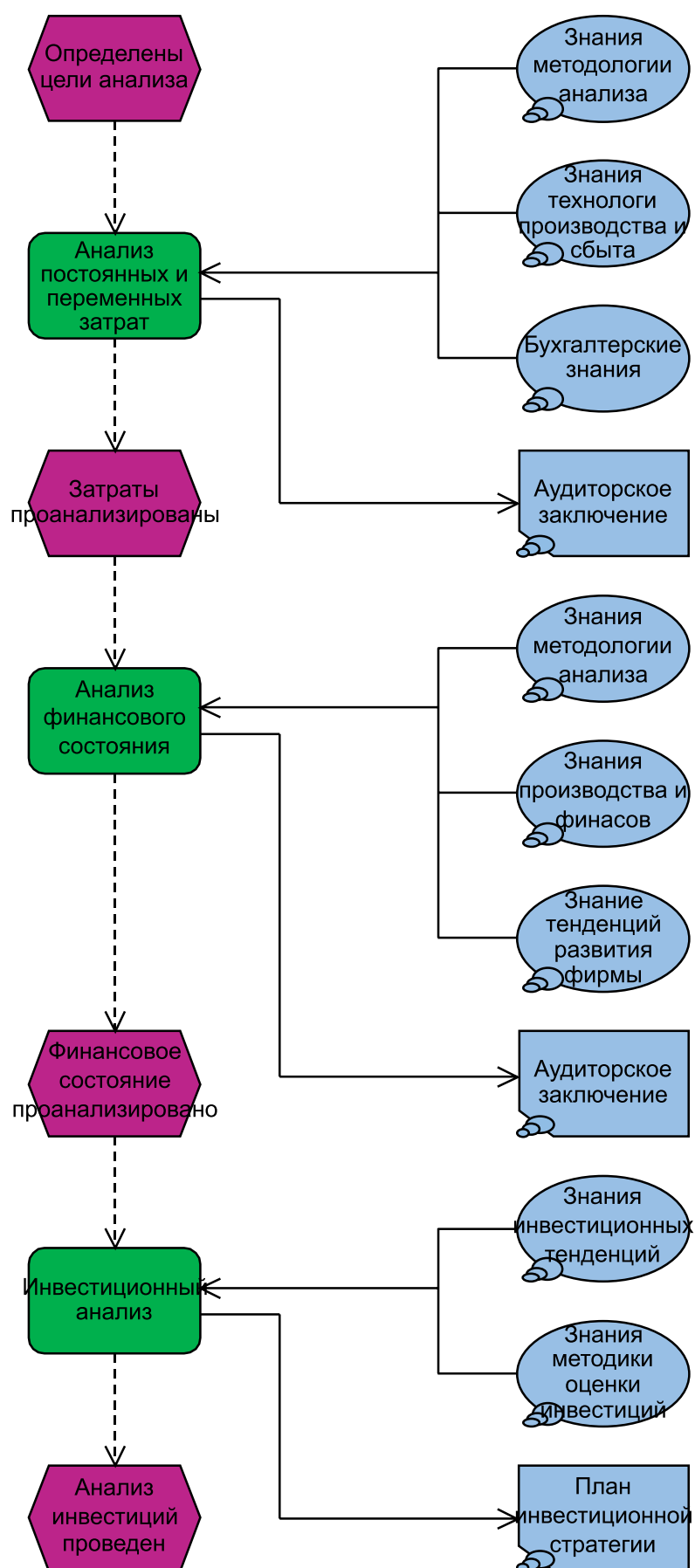


Рис. 6.7.4 Обработка знаний в eEPC

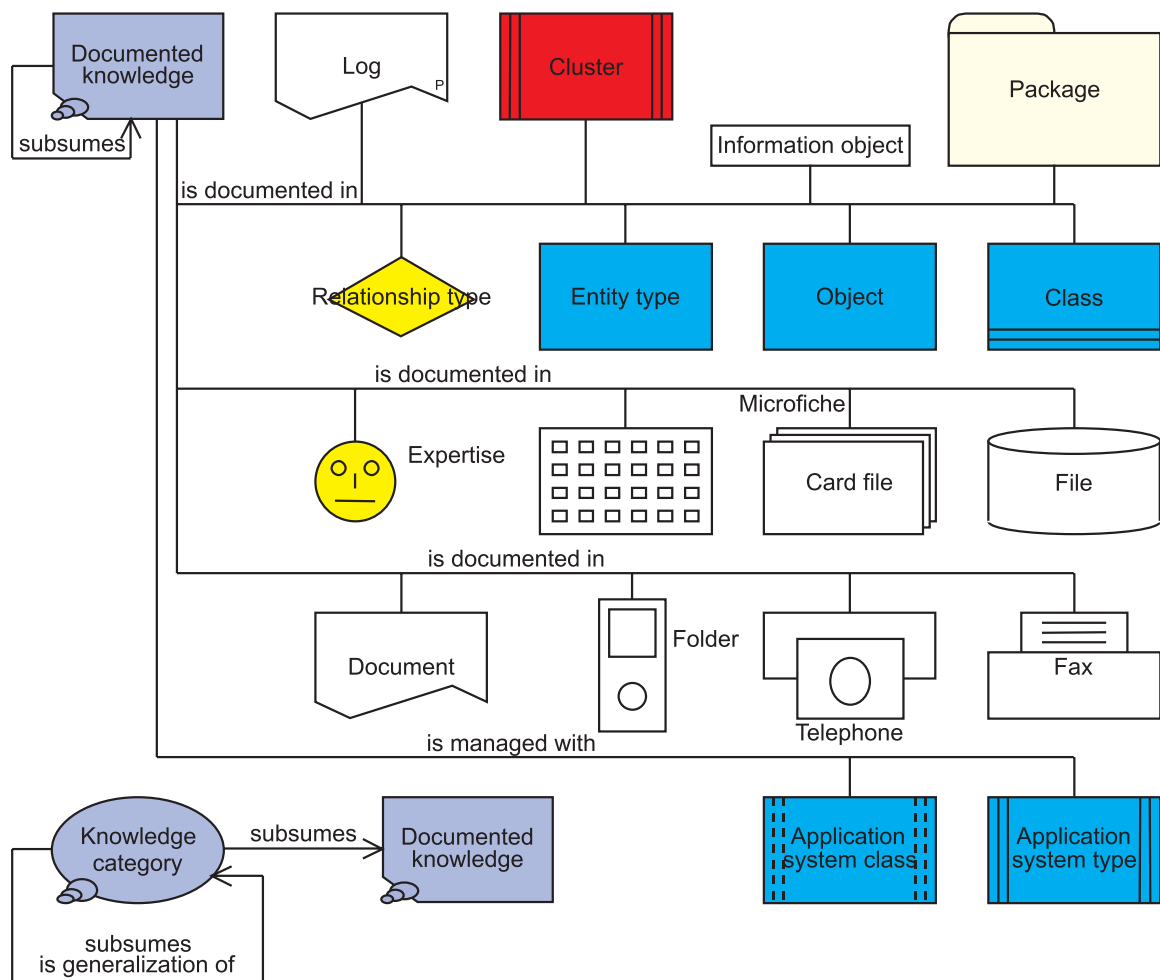
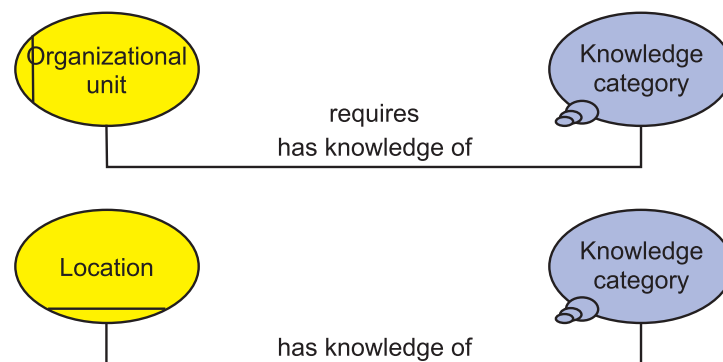


Рис. 6.7.5 Объекты и связи модели типа Knowledge structure diagram



Здесь на месте объекта Organizational unit может стоять любой из следующих объектов

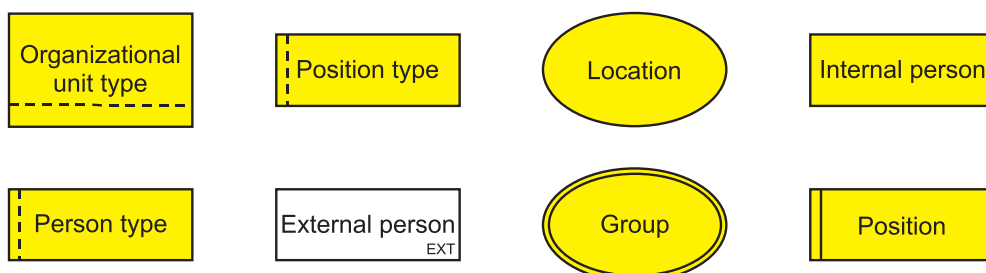


Рис. 6.7.6 Объекты и связи модели типа Knowledge map (Карта знаний)

системного интерфейса. *Типы диаграмм* – диаграмма цепочки добавленного качества, организационная диаграмма, eEPC, диаграмма прецедентов, диаграмма прикладных систем, диаграмма типов прикладных систем, блок-схема программы, диаграмма экранов.

*Задача – управление знаниями. Подзадачи* – подготовка карты знаний, классификация знаний, обработка знаний в БП. *Типы диаграмм* – карта знаний, структурная диаграмма знаний, eEPC, PCD, офисный БП, производственный БП, диаграмма распределения функций.

*Задача – внедрение системы управления потоками работ. Подзадачи* – описание БП (офисных и промышленных), реализация системы управления БП. *Типы диаграмм* – диаграмма прав, eEPC, диаграмма распределения (окружения) функций, диаграмма прикладных систем, диаграмма типов прикладных систем.

После построения моделей можно использовать еще одну функциональность ARIS Toolset – анимацию, которая особенно удобна для процессно-ориентированного обучения. При использовании анимации модель БП можно представить в динамике и проанализировать такие параметры БП, как время и стоимость выполнения. При этом могут вызываться присвоенные документы, приложения и Intranet/Internet связи. Процесс анимации может быть задокументирован как бизнес-случай и сохранен под собственным именем.

Анимация и бизнес-случаи используются для следующих целей: просмотра динамической логики БП, упрощения процесса моделирования логически сложных БП, анализа сроков и затрат для одного, нескольких или всех БП производственной деятельности предприятия, оценки минимальных (максимальных) материальных затрат и времени исполнения различных альтернативных БП, визуализации интегрированных документов, прикладных систем и Intranet/Internet страниц в БП.

Анимации могут быть подвергнуты все модели БП типа eEPC, eEPC с материальным потоком, производственных БП, офисных БП, PCD и PCD с материальным потоком, в которых рассматриваются функции, события, правила, связи и типы объектов интерфейса процесса.

Анимация (с регистрацией или без) может быть осуществлена с *ручным контролем*, где каждый анимационный шаг инициируется пользователем, и с *полуавтоматическим контролем*, где несколько шагов в БП выполняются системой автоматически. Если есть ответвления, система спрашивает, какой путь будет принят.

В процессе анимации пересчитываются следующие параметры: число функций, событий, соединений, интерфейсов процесса, запущенных приложений, среднее время ожидания, общая стоимость, матери-

альные затраты, затраты на персонал и др. Эти данные выводятся в отдельном окне и непрерывно подсчитываются на каждом анимационном шаге.

Документы и приложения, присвоенные процессно-релевантным объектам, могут быть выполнены в анимации. Для этого есть следующие базовые опции. Во-первых, *автоматическое выполнение приложений*. Как только объект процесса активизирован, то выполняется связанный с ним документ или приложение. Во-вторых, *автоматический запуск с анимационной остановкой*. Как только объект процесса активизирован, то анимация приостанавливается и выполняется документ или приложение, которые связаны с объектом. Анимация останавливается до тех пор, пока приложение открыто. В-третьих, *запуск с запросом*. Если документ или приложение присвоены объекту, то пользователя спрашивают, следует ли их выполнять. В-четвертых, *запуск вручную*, где документы и приложения могут быть запущены только вручную.

Для выполнения сохраненных бизнес-случаев имеются следующие возможности. Во-первых, *автоматическое выполнение*, где предварительно зарегистрированный бизнес-случай выполняется автоматически и во время выполнения его можно остановить. Во-вторых, *пошаговое выполнение*, где предварительно зарегистрированный бизнес-случай выполняется по шагам и каждый отдельный шаг должен быть инициирован пользователем.

На основе разработанных моделей можно создать набор документов, в котором отражена проведенная работа. Эти документы должны служить исходным материалом для дальнейших этапов моделирования, тестирования, анализа, оптимизации и использования моделей для других целей. Отчет по проделанной работе создается при помощи модуля ARIS Report. Ниже перечислены и кратко описаны скрипты (сценарии) отчетов по созданным моделям.

*Activation\_R3.rsm*. Выводит информацию о состоянии активации объектов заданных типов (функций, событий организационных единиц). Обрабатывается 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.

*Assignments\_R3.rsm*. По каждой модели, имеющей отношение к системе SAP R/3, выводит перечень функций, для которых имеется детализация. Обрабатывается информация о коде транзакции, абсолютном пути к детализирующей модели, типе детализирующей модели, символе анализируемого объекта и статусе его активизации.

*Aris2xmiRR.rsm (Aris2xmiRR\_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. 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 скрипте.

*CBusinessScenarioAggregated.rsm.* Выводит общую информацию об участниках бизнеса и их деловых процессах с отображением входов и выходов. Отчет создается для моделей типа “Диаграмма сценариев е-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 е-бизнеса”.

*Function\_r3.rsm.* Используется для обзора функций, имеющих отношение к системе SAP R/3. Содержит информацию о коде транзакции, критерии классификации, состоянии активации. Возможен вывод графики модели.

*FunctionSelection\_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.

*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 приложе-

ния. Пользователем задается формат вывода (ориентация листа, его размер и поля, правила разделения модели на страницы, коэффициент масштабирования).

*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.

*ModelInfo.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.

*ModelIso.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.* 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.* 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 и 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.* Описывает атрибуты и графические символы диаграммы цепочек добавленного качества. 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.

Стандартные скрипты модуля *ARIS Report* располагаются в каталоге *ARIS5\Script\Report\en*. Сгенерированные отчеты по умолчанию располагаются в каталоге *ARIS5\Script\Report\out*.

Отчеты можно создавать также и для других ARIS-сущностей. Приведем их названия и краткие описания.

### Скрипты отчетов по объектам

*FunctionalHierarchy.rso*. Выводит в отчет иерархию функций в БД. Для выбранного объекта типа *функция* фиксируются все связанные с ней функции, содержащиеся в БД. 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.

*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.

*Object\_3x.rso*. Выводит информацию о папках, моделях и находящихся в них выделенных объектах и их связях.

*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 информацию о местона-

хождении модели в БД.

*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 – по алфавиту.

*OrgUnit\_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 – по алфавиту.

*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.

### Скрипты отчетов по БД

*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\_Data Export.rsm* и изменены в соответствии с текущим состоянием стратегии компании.

*DBLanguagesFontFonnats.rsd*. Выводит информацию о всех используемых в БД языках и форматах шрифтов.

*DBUser.rsd*. Описывает всех пользователей и их группы для выбранной БД и выдает информацию относительно назначенных для них функциональных привилегий.

*IDCheck.rsd*. Осуществляет проверку идентификаторов всех элементов БД 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 БД с одного языка на другой. Осуществляет перенос в ARIS информации, предварительно переведенной в системе Excel.

### Скрипты отчетов по папкам БД

*CMA\_ActivUies.rsg*. Формирует список действий по планируемым изменениям в моделях и объектах, 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.

*GroupInfo.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.

*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 БД 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 этого скрипта.

### Скрипты отчетов по таблицам функционально-стоимостного анализа

*CostCenterAnalysis\_\_1.rst*. Выводит результаты анализа затратных центров, выполненного методом ABC.

*ProcessCalculation.rst*. Выводит результаты расчета стоимости процессов, произведенных методом ABC.

### Скрипт отчета по фильтрам

*FilterInfo.rsf*. Выводит информацию о входящих в установленный методологический фильтр моделях, объектах, их атрибутах, разрешенных связях и т. д. Пользователь может выбирать нужную ему информацию. Вывод осуществляется в выбранном виде, например, текстовом или табличном.

## 6.9. Семантический контроль

При моделировании необходимо соблюдать *синтаксические* и *семантические* правила. Синтаксический контроль проводится в процессе моделирования, где ARIS-система сама не позволяет выполнять запрещенные действия, например, использовать запрещенные объекты, связи, виды детализации и т. д.

Семантические проверки предназначены для обеспечения целостности и непротиворечивости моделей БП. Они представляют собой особую форму отчета. В результате семантической проверки правильность и законченность моде-

ли может быть оценена на основе выходных данных, которые показывают число отклонений от правил моделирования и перечисляют конкретные нарушения. В этот перечень входят модели (с указанием пути загрузки, типа и имени модели), правила, которые не соблюдались, и объекты, в которых были нарушены правила (с указанием типа объекта и его имени). Все это позволяет осуществить проверку моделей на предмет соблюдения установленных пользователями соглашений. Полученный отчет можно использовать для принятия решения о завершении процесса построения модели.

Контроль ARIS-сущностей осуществляется с помощью скриптов. *Скрипт* – это программа (макрос) на языке ARIS-Basic, которая обрабатывает сценарий контроля. Каждая сущность имеет доступ к определенному типу скрипта. Две первые буквы расширения скрипфайла определяют тип скрипта. Например, компоненты ARIS Report, ARIS Analysis и ARIS Semantic Check имеют доступ к файлам с расширением *rs\**, *as\** и *se\** соответственно. В каждом типе скрипта различают подтипы, обозначаемые третьей буквой расширения файла. В таблице 6.9.1 представлены расширения подтипов.

Таблица 6.9.1

| Подтип                          | Расширение файла |
|---------------------------------|------------------|
| Общий                           | **c              |
| Ориентированный на объект       | **o              |
| Ориентированный на модель       | **m              |
| Ориентированный на группу       | **g              |
| Ориентированный на пользователя | **u              |
| Ориентированный на фильтр       | **f              |
| Ориентированный на БД           | **d              |
| Ориентированный на таблицу      | **t              |

Для проведения семантической проверки моделей и объектов компонента ARIS Semantic Check имеет доступ к следующим подтипам правил проверки – *Allocation-Rules.sem* (Правила взаимосвязи объектов), *Assignment-Rules.seo* (Правила детализации объектов), *Existence-Rules.sem* (Правила существования объектов), *ObjectAttributeRules.sem* (Правила атрибутов объектов в моделях), *ObjectAttributeRules.seo* (Правила атрибутов объектов), *RelationshipAttributeRules.sem* (Правила атрибутов связей), *StructureRules.sem* (Правила структуры моделей), *UMLRules.sem* (Правила универсального языка моделирования).

Перечислим некоторые правила.

*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* проверяют корректность формирования структуры взаимосвязей объектов в модели. Эти 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 «X» или «V», в модели запрещены циклы.

*Специальные структурные правила для иерархических моделей* (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 атрибутов для метода ABC.

*Правила детализации* 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 дета-

лизована в виде дерева функции, функция должна быть детализирована в виде диаграммы окружения функций.

Существуют расширяемые правила проверки, которые легко изменять, используя закладку *Semantic Check* (Семантическая проверка), вызываемую через пункт меню *View/Option*. К ним относятся правила существования объектов, взаимосвязей объектов, атрибутов объектов, атрибутов связей объектов и UML.

Методику проведения семантической проверки рассмотрим на примере модели eEPC, изображенной на рис. 6.9.1 и содержащей ряд семантических ошибок. На рис. 6.9.2 эта же модель изображена после устранения ошибок.

Для выполнения проверки выбранной модели в меню *Evaluate* (Оценка) выбираем команду *Semantic Check* (Семантическая проверка) (рис. 6.9.3). На экране появляется ДО *Semantic Check Wizard – Select Rule Type* (Мастер семантической проверки – Выбор типа правила) (рис. 6.9.4). В поле ввода с раскрывающимся списком *Suggested Semantic Check Scripts (Rule Types)* (Предлагаемый сценарий семантической проверки) имеется шесть типов правил: *AllocationRules.sem*, *ExistenceRules.sem*, *ObjectAttributeRules.sem*, *RelationshipAttributeRules.sem*, *UMLRules.sem*, *StructureRules.sem*.

Выбираем тип правила *StructureRules.sem*, на основе которого будет проведена семантическая проверка. При этом в левой части ДО появляется наименование выбранного типа правила и его описание. Поле *Other Script* (Другие сценарии) предназначено для использования сценариев, которые разработаны пользователем. После нажатия кнопки «Далее» появляется ДО *Semantic Check Wizard – Select Output Options* (Мастер семантической проверки – Выбор опций представления результатов) с предложением выбрать опции представления результатов (рис. 6.9.5).

После нажатия на кнопку «Готово» появляется окно *ARIS Semantic Check* (Семантическая проверка ARIS) (рис. 6.9.6). В поле *Rule Groups* (Группы правил) выбираем группу правил. К выбранному типу правил относятся пять групп правил:

General structure rules for all model types – Общие структурные правила для всех типов моделей;

Structure rules for process models – Структурные правила для процессных моделей;

Structure rules for eERMs – Структурные правила для eERM-моделей;

Structure rules for hierarchy models (function trees, organizational charts, etc.) – Структурные правила для иерархических моделей (функциональных деревьев, организационных диаграмм, и т. д.);

Structure rules for specified hierarchy models – Структурные правила для определенных иерархических моделей.

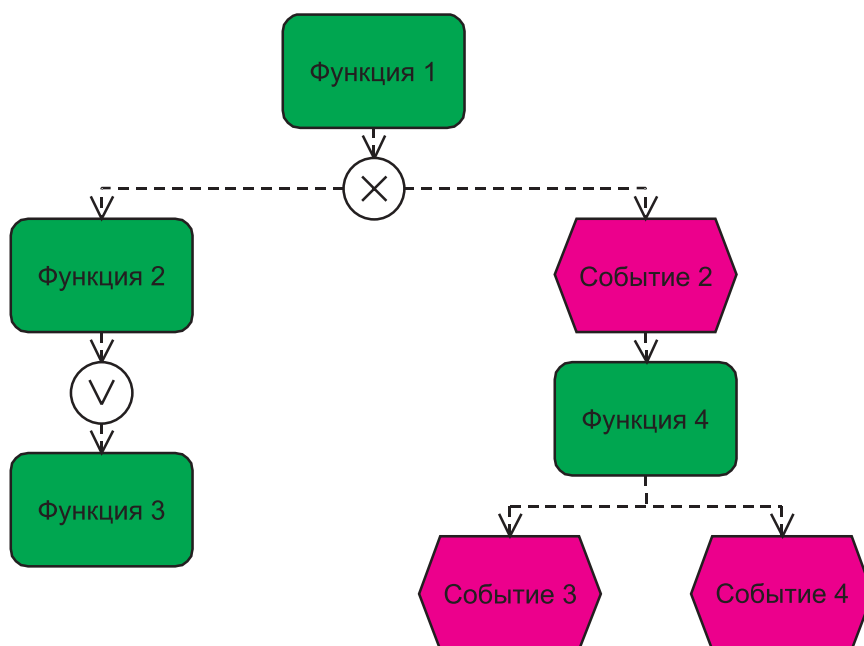


Рис. 6.9.1 Первый пример с ошибками

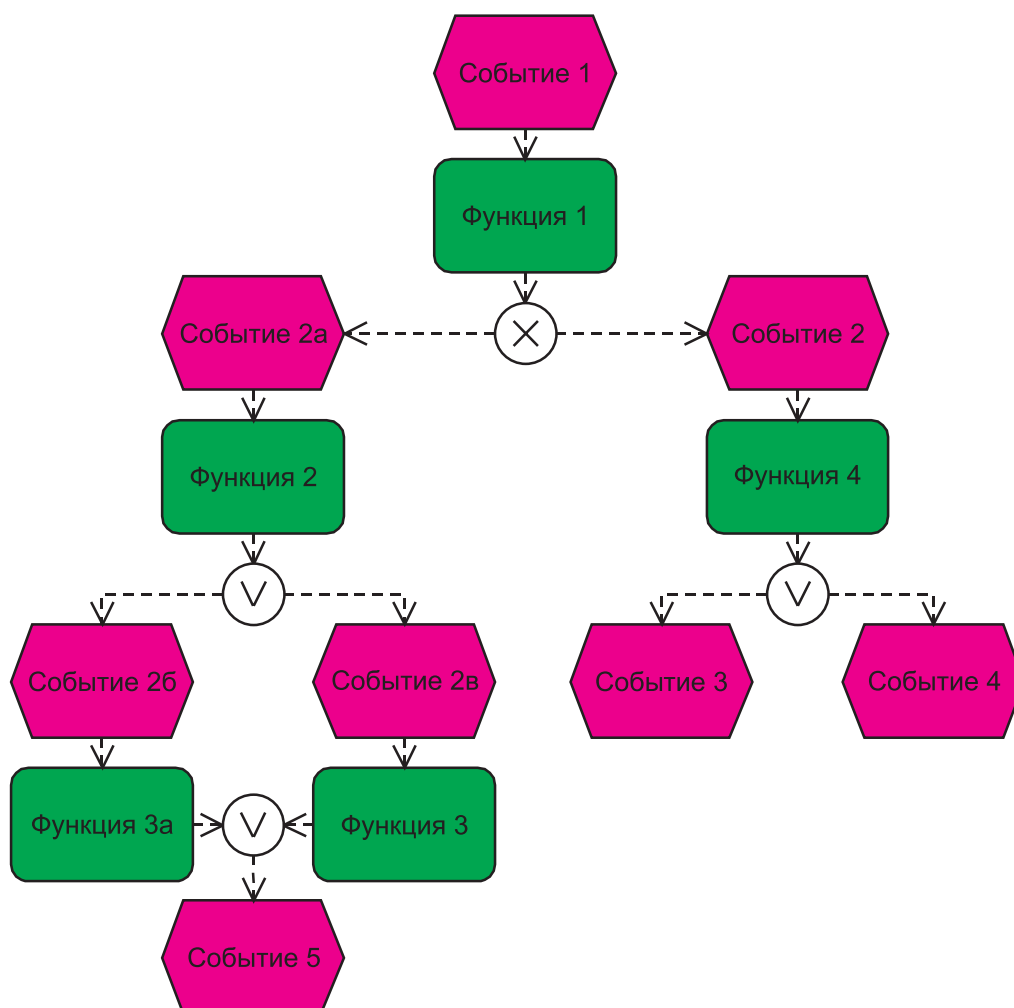


Рис. 6.9.2 Первый пример после устранения ошибок

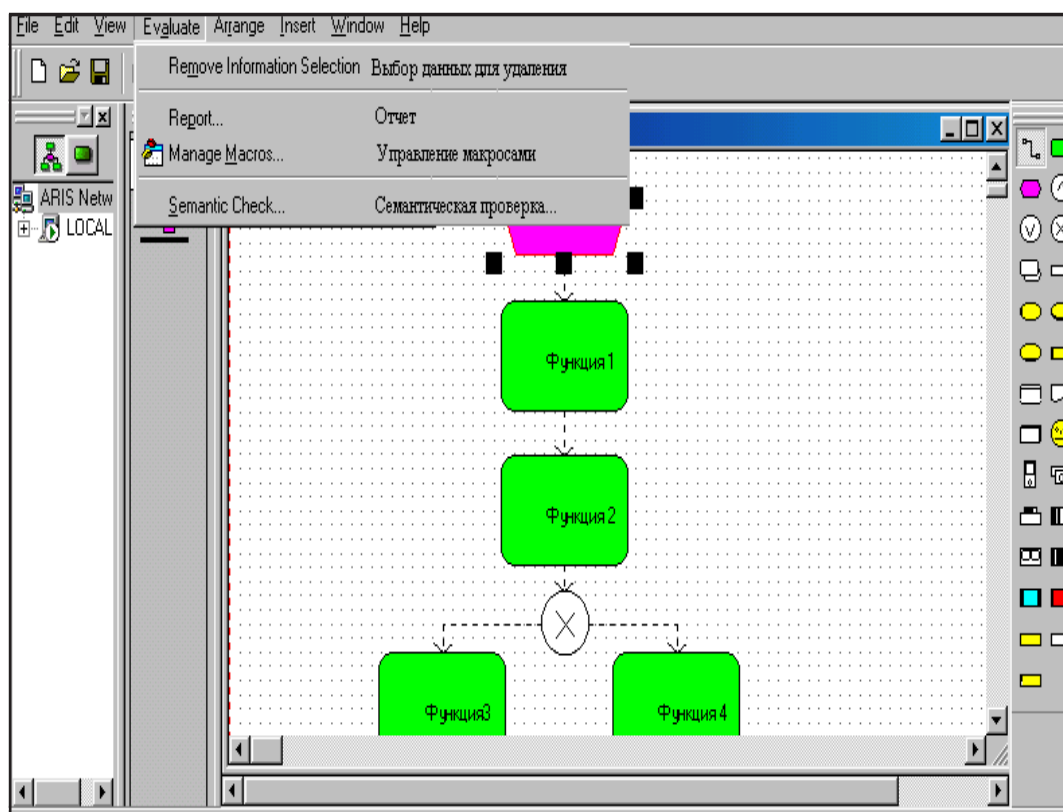


Рис. 6.9.3 Выбор пункта меню “Семантическая проверка”

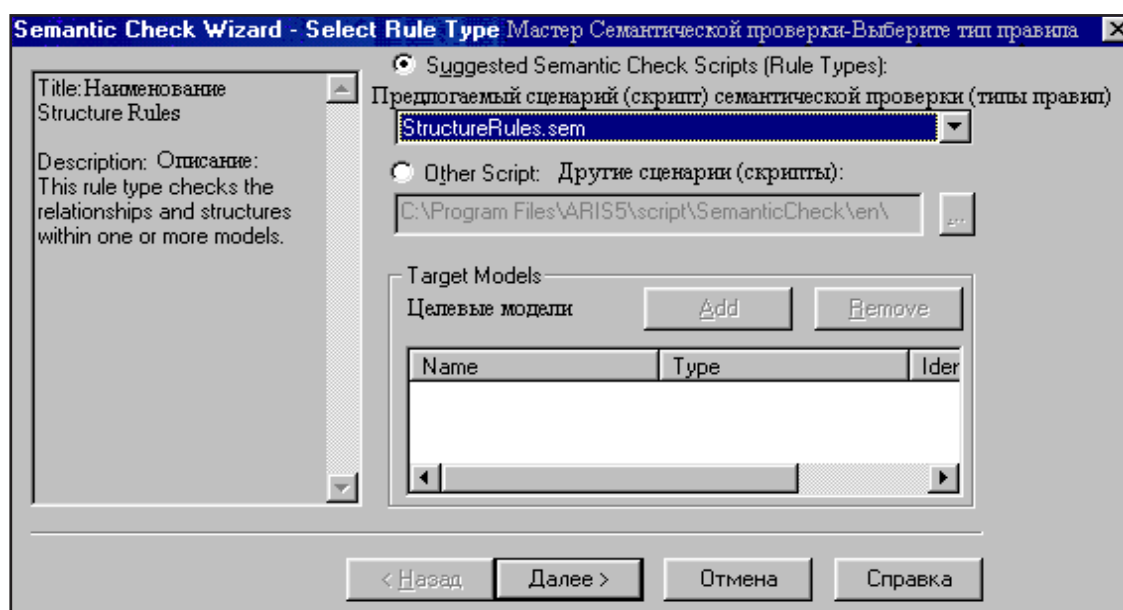


Рис. 6.9.4 Диалоговое окно выбора типа правила



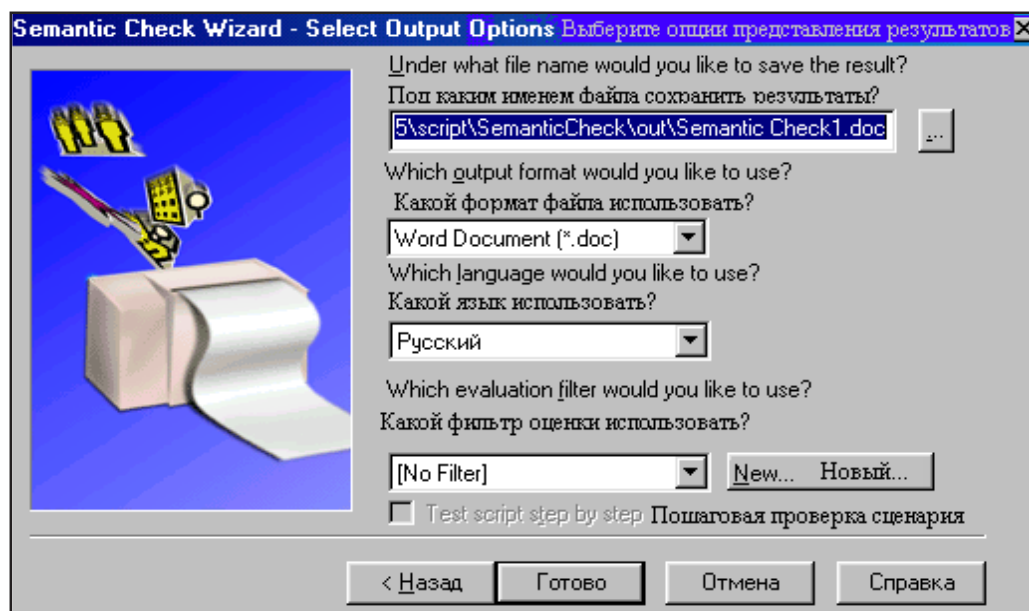


Рис. 6.9.5 Диалоговое окно выбора опций представления результатов

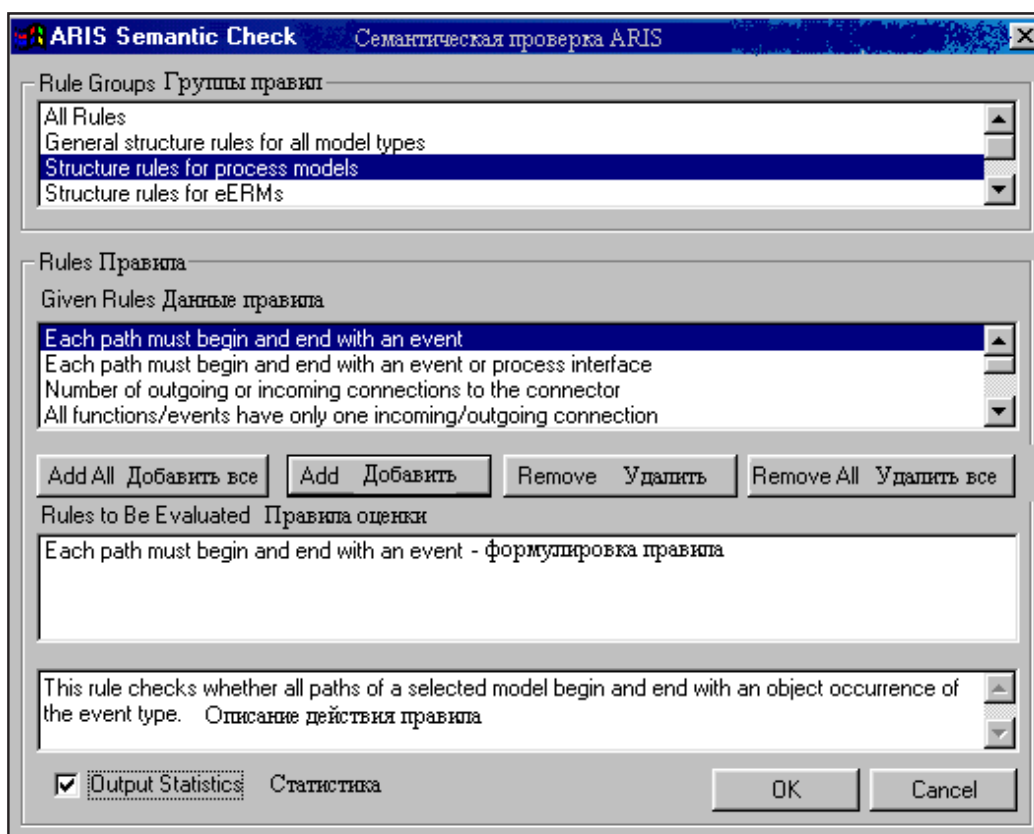


Рис. 6.9.6 Окно семантической проверки



Выбираем группу *Structure rules for process models* (Структурные правила для процессных моделей). При этом в поле *Given Rules* (Данные правила) группы Rules (Правила) появляются правила, входящие в выбранную группу. Можно выбрать все правила для проверки, нажав кнопку "Add All" ("Добавить все"), несколько правил или конкретное правило, используя кнопку "Add" ("Добавить"). При этом правила, выбранные для проверки, появляются в поле *Rules to Be Evaluated* (Правила оценки). При установке флага "Output Statistics" ("Статистика") в файле отчета будет содержаться дополнительная статистическая информация о количестве ошибок по каждому проверяемому правилу, а также имя и тип проверяемых моделей.

Выбираем правила: *Each path must begin and end with an event* (Каждый путь должен начинаться и заканчиваться событием); *Number of Outgoing and incoming Connections at the operator* (Количество входящих и исходящих связей оператора); *All functions/events have only one incoming/outgoing connection* (Все функции или события имеют только одну входящую и исходящую связь); *No XOR/OR Possible after Event* (Невозможность XOR/OR после события); *Sequence at the Operator Needs To Be Followed* (Соблюдение последовательности при использовании оператора).

Нажимаем кнопку "OK". На экране появляется окно *Semantic Check* (Семантическая проверка) и начинает формироваться отчет проверки (рис.6.9.7). Нажав кнопку "Cancel", можно прервать этот процесс. После завершения формирования отчета появляется окно "Преобразование файла", где выбирается формат файла, в котором будут храниться результаты отчета (рис.6.9.8). Когда процесс формирования отчета заканчивается, на экране появляется сообщение об успешном выполнении сценария, изображенное на рис.6.9.9. После нажатия на кнопку "Да" загрузится редактор "Winword" с файлом отчета.

Отчет содержит имя, описание правила и таблицу из трех колонок: в первой колонке перечислены проверенные модели; во второй колонке – объекты, в которых были нарушены правила; в третьей колонке указываются типы объектов, в которых нарушены правила. Сортировка в таблице осуществляется по модели, по типу объекта и по имени объекта.

Статистическая часть отчета содержит информацию о количестве допущенных при моделировании ошибок. Она формируется также в виде таблицы, состоящей из двух колонок: в первой перечислены по порядку все правила, выбранные для проверки, а во второй количество ошибок на правило.

В результате проверки процесса "Первый пример" получаем отчет, приведенный в приложении 4.

В таблице 6.9.3 представлена статистическая часть отчета.

Таблица 6.9.3

| <b>Rule(правило) / Models № (модель №)</b>                            | <b>1</b> |
|---|----------|
| 1. Каждый путь должен начинаться и заканчиваться событием             | 2        |
| 2. Количество входящих и исходящих связей оператора                   | 3        |
| 3. Все функции и события имеют только одну входящую и исходящую связь | 1        |
| 4. Невозможность XOR/OR после события                                 | 0        |
| 5. Должен сохраняться порядок при операторе                           | 2        |

Видим, что при моделировании мы допустили восемь ошибок. Устранив выявленные ошибки, получим модель, изображенную на рис.6.9.2.

В качестве 2-го примера выполним семантическую проверку организационной диаграммы, изображенной на рис.6.9.10 с ошибками и на рис.6.9.11 после их устранения.

Для проверки выбираем следующие правила: *Only one root possible* (Возможен только один корень); *Each object may have only one direct predecessor* (Каждый объект может иметь только одного непосредственного предшественника); *All connections in a model must be of the same type* (Все соединения в модели должны быть одного типа).

Отчет этой проверки содержится в приложении 5. Статистическая часть отчета проверки содержится в таблице 6.9.4.

Таблица 6.9.4

| <b>Rule(правило)/Models № (модель №)</b>                                     | <b>1</b> |
|--|----------|
| 1. Возможен только один корень   | 0        |
| 2. Каждый объект может иметь только одного непосредственного предшественника | 1        |
| 3. Все соединения в модели должны быть одного типа                           | 5        |

Анализируя данные полученного отчета, видим, что по второму правилу найдена одна ошибка. Сотрудник №2 первого отдела является подчиненным для начальников обоих отделов, что противоречит наименованию его должности, т. е. он является подчиненным только для начальника первого отдела. Связь между этим сотрудником и начальником отдела №2 необходимо удалить.

По третьему правилу найдено пять ошибок. Одну мы уже исправили, устранив предыдущую ошибку. В проверяемом

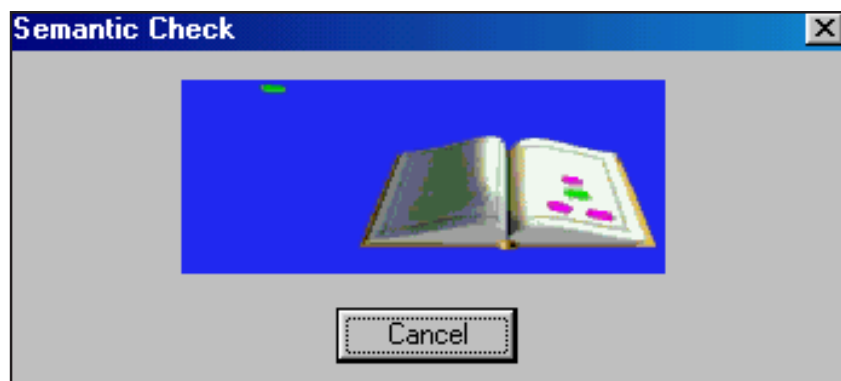


Рис. 6.9.7 Окно процесса семантической проверки

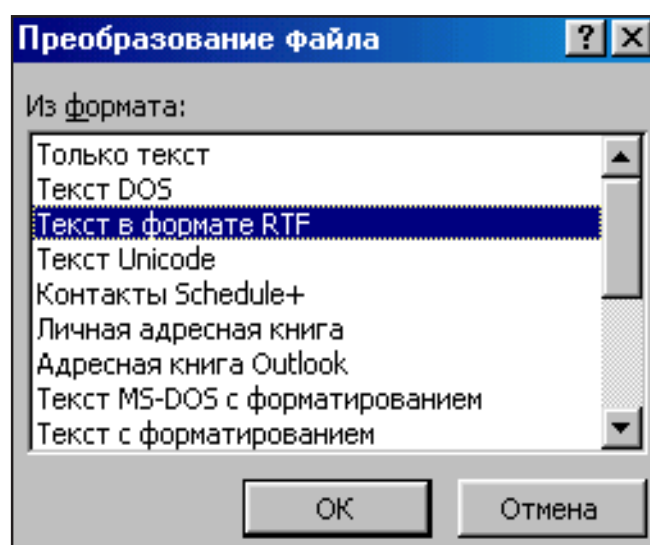


Рис. 6.9.8 Окно выбора файла для результатов отчета

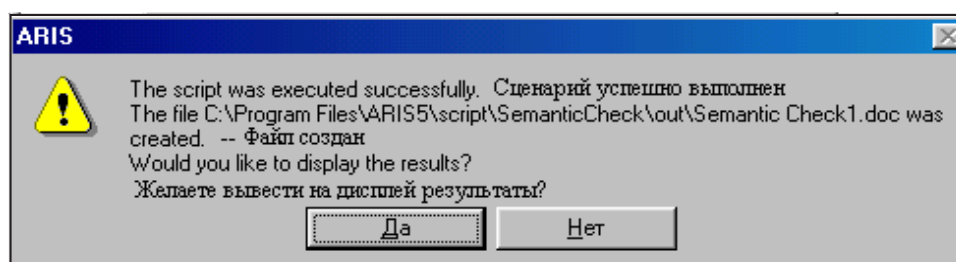


Рис. 6.9.9 Окно информации завершения проверки

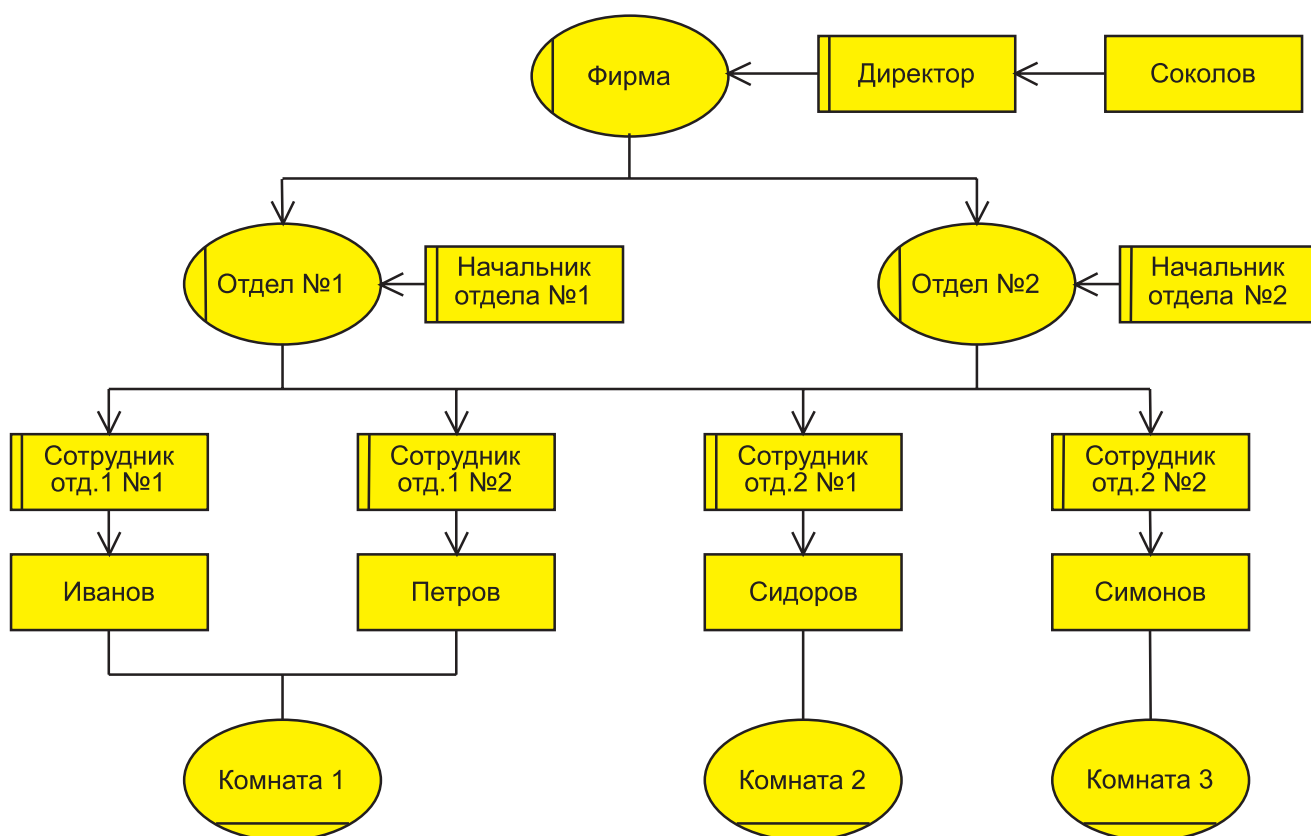


Рис. 6.9.10 Второй пример с ошибками

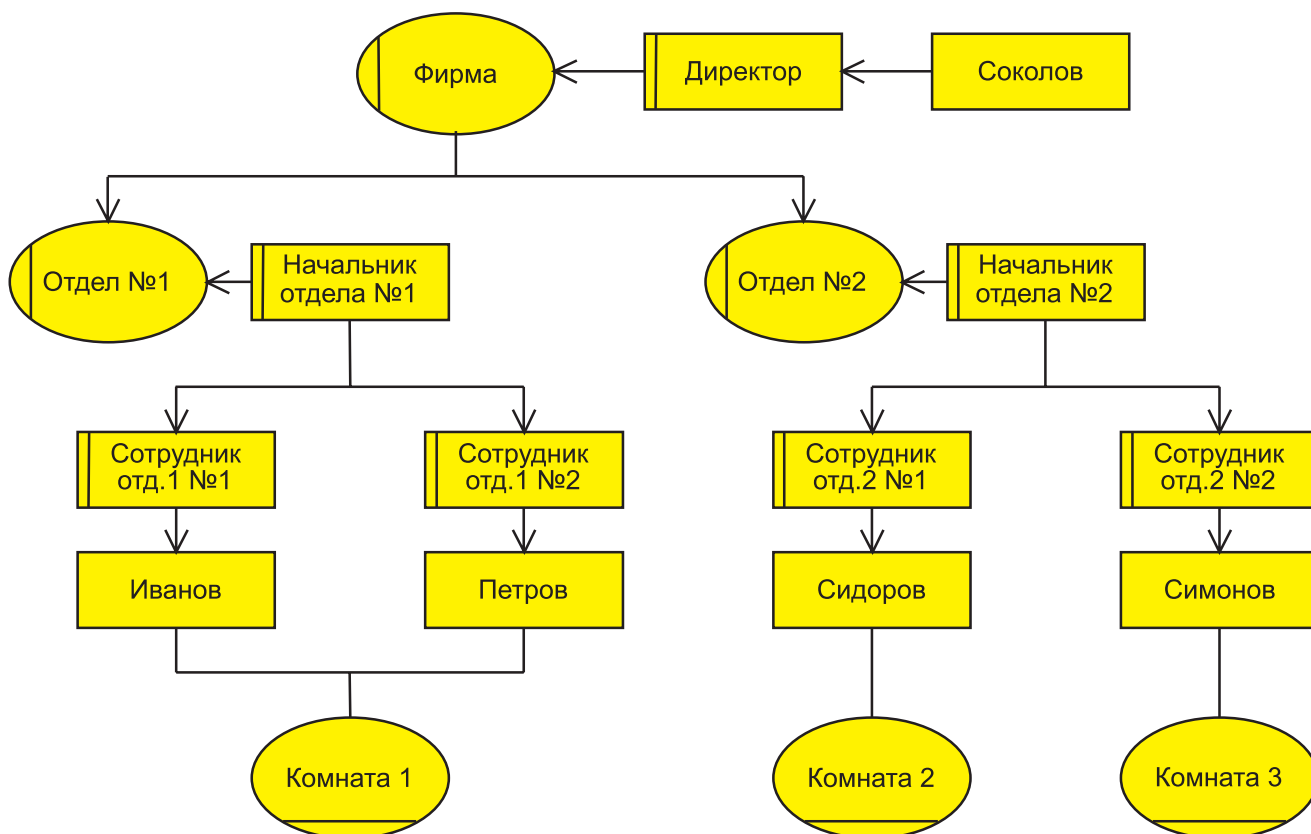


Рис. 6.9.11 Второй пример после устранения ошибок

примере тип связи “сформирована” между отделами и сотрудниками является неуместным. Анализируя модель, видим, что фирма состоит из отделов. Во главе фирмы и отделов соответственно стоят директор и начальники отделов. Отделы входят в состав фирмы. Значит, начальники отделов подчиняются директору. Поэтому следует заменить ошибочные связи между объектами “отдел – сотрудник” на связи между объектами “начальник – сотрудник”, т. е. вместо типа связи “сформирован” использовать тип связи “является менеджером для”.

Доработанная модель изображена на рис.6.9.11. Проведем повторную проверку, используя только те правила, в которых были допущены ошибки. Статистическая часть отчета содержится в таблице 6.9.5.

Таблица 6.9.5

| <b>Rule(правило) / Models № (модель №)</b>                                   | <b>1</b> |
|--|----------|
| 1. Каждый объект может иметь только одного непосредственного предшественника | 0        |
| 2. Все соединения в модели должны быть одного типа                           | 0        |

Видим, что ошибок на выбранные правила нет.

В качестве следующего примера выполним семантическую проверку функционального дерева, изображенного на рис.6.9.12 с ошибками и на рис.6.9.13 после их устранения.

Выбираем правила: *Objects without Connections* (Объекты без связей); *Connections between Two Different Objects* (Связи между двумя различными объектами); *No cycles may exist in the model* (В модели не могут существовать циклы); *Each object may have only one direct predecessor* (Каждый объект может иметь только одного непосредственного предшественника).

Отчет проверки содержится в приложении 6. Статистическая часть отчета содержится в таблице 6.9.6.

Таблица 6.9.6

| <b>Rule(правило) / Models № (модель №)</b>                                   | <b>1</b> |
|--|----------|
| 1. Объекты без связей  | 1        |
| 2. Связи между двумя различными объектами                                    | 1        |
| 3. В модели не могут существовать циклы                                      | 1        |
| 4. Каждый объект может иметь только одного непосредственного предшественника | 1        |

При моделировании допущено по одной ошибке на каждое выбранное правило. Видим, что “Функция 8” не имеет

связи, значит, она либо лишняя, либо должна иметь связь с каким-нибудь объектом. Появление одного и того же объекта “Функция 2” на двух уровнях иерархии невозможно. Использование цикла делает непонятным механизм функционирования модели. “Функция 5”, согласно описанию дерева функций, не может иметь двух непосредственных предшественников.

Устранив выявленные недостатки, получим модель, изображенную на рис.6.9.13.

Проведем повторную проверку, используя только те правила, в которых были допущены ошибки. Статистическая часть отчета по результатам этой проверки содержится в таблице 6.9.7.

Таблица 6.9.7

| <b>Rule (правило) / Models № (модель №)</b>                                  | <b>1</b> |
|--|----------|
| 1. Объекты без связей  | 0        |
| 2. Связи между двумя различными объектами                                    | 0        |
| 3. В модели не могут существовать циклы                                      | 0        |
| 4. Каждый объект может иметь только одного непосредственного предшественника | 0        |

Видим, что ошибок на выбранные правила нет.

В последнем примере выполним семантическую проверку модели технических терминов, изображенную на рис.6.9.14 с ошибками и на рис.6.9.15 после их устранения.

Выбираем правила: *Objects without Connections* (Объекты без связей); *Connections between Two Different Objects* (Связи между двумя различными объектами).

Отчет проверки содержится в приложении 7. Статистическая часть отчета содержится в таблице 6.9.8.

Таблица 6.9.8

| <b>Rule (правило) / Models № (модель №)</b> | <b>1</b> |
|---|----------|
| 1. Объекты без связей                       | 1        |
| 2. Связи между двумя различными объектами   | 0        |

При моделировании допущена одна ошибка на первое правило. Видим, что объект “Кластер 3” не имеет связи, значит, он либо лишний, либо должен иметь связь с каким-нибудь объектом.

Устранив ошибку, получим модель, изображенную на рис.6.9.15.

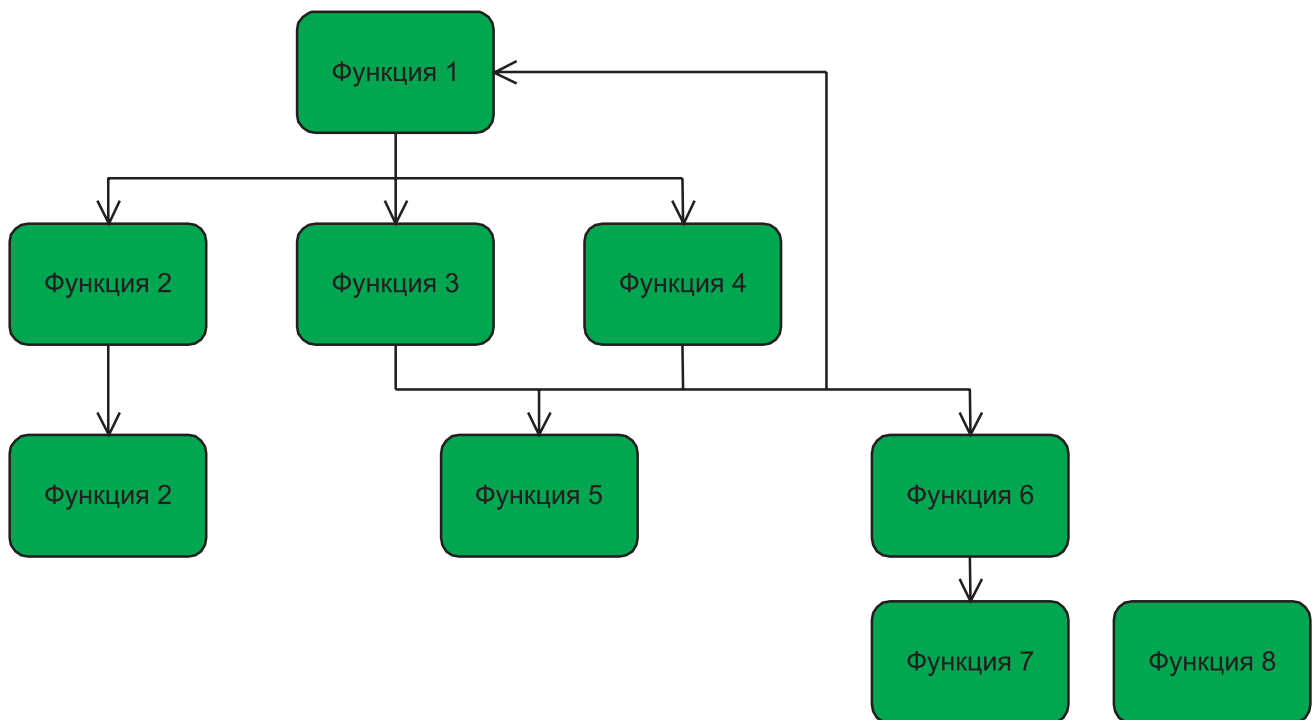


Рис. 6.9.12 Третий пример с ошибками

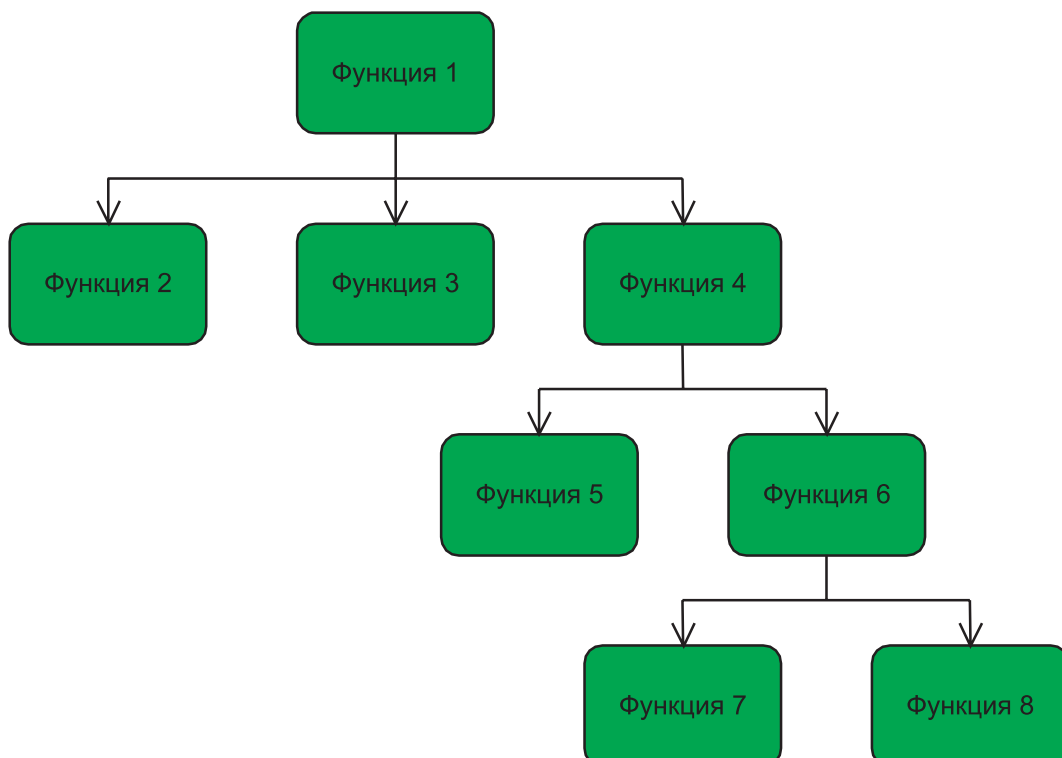


Рис. 6.9.13 Третий пример после устранения ошибок

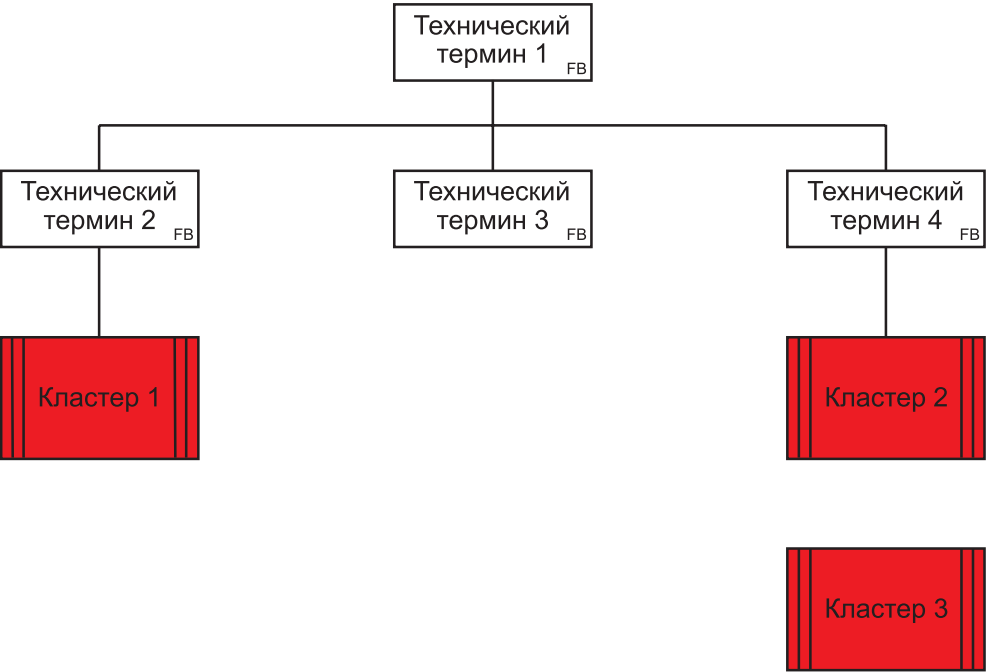


Рис. 6.9.14 Четвертый пример с ошибками

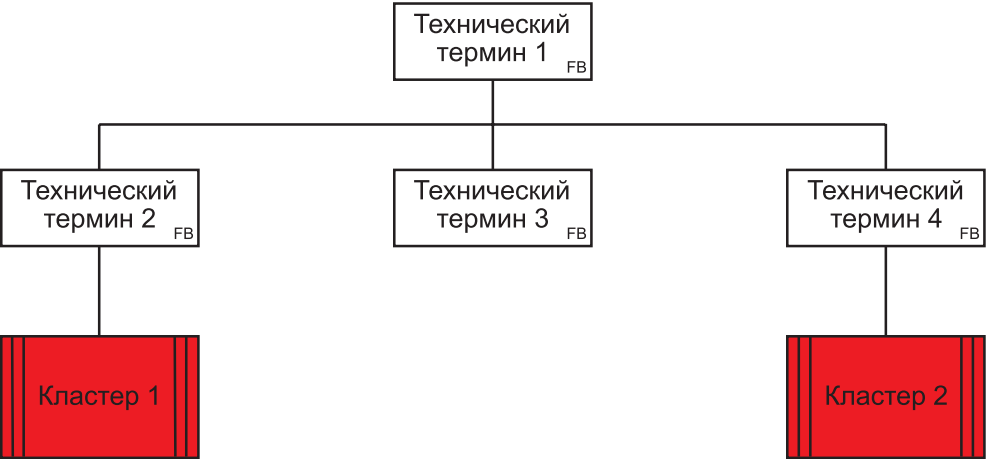


Рис. 6.9.15 Четвертый пример после устранения ошибок



## 7. ARIS-МОДЕЛИРОВАНИЕ С ЛЕГКИМ ФИЛЬТРОМ

ARIS 5.0 с легким фильтром (Easy Filter) предоставляет пользователю всего восемь типов моделей (рис.6.1.3). В этой главе достаточно подробно рассмотрены эти типы моделей, в том числе их атрибуты, а также атрибуты всех допустимых типов объектов и связей между ними. Для каждого вида моделирования представлены примеры ARIS-программ (скриптов) анализа моделей. Эти программы достаточно подробно комментированы, и для использованных методов ARIS-объектов приведены краткие описания.

### 7.1. Описание атрибутов моделей, объектов и связей

Все типы моделей, объектов и связей в ARIS 5.0 с Easy Filter, за редким исключением, имеют одинаковые атрибуты. Например, каждый из всех девятнадцати объектов имеют по тринадцать одинаковых атрибутов, и только три объекта имеют дополнительные атрибуты (от одного до восьми). Это же можно сказать об атрибутах моделей и связей между объектами. Перейдем к описанию типов объектов, связей и их атрибутов.

В таблице 7.1.1 перечислены атрибуты моделей, допустимых в ARIS 5.0 с Easy Filter. Здесь приведены их английские и русские названия и кратко описано их назначение.

В таблице 7.1.2 представлены все типы атрибутов, которые описывают экземпляры допустимых объектов. Здесь даны их названия и кратко описано назначение.

Из этой таблицы видно, что все объекты восьми моделей в главной папке (группе Attributes) содержат семь атрибутов: Name, Identifier, Description/Definition, Full name, Remark/Example, Type, Creator. Объект Person в группе Attributes имеет следующие дополнительные атрибуты: Telephone number, Fax number, E-mail address, Address, которые позволяют описать специфику персон (личностей). Подгруппа (вложенная папка) System attributes содержит шесть атрибутов: Title 1, Link 1, Parameter 1, Title 2, Link 2, Parameter 2. Кроме этого, объект Organization unit содержит дополнительную подгруппу (новую папку) Simulation с единственным атрибутом Number of employees (число служащих). Две дополнительные подгруппы имеет объект типа Function, где подгруппа Costs (цена, затраты) содержит папку Material costs с атрибутом Avg. material costs (средние материальные затраты), папку Other costs (прочие или другие затраты) с атрибутом Avg. other costs, папку Personnel costs с атрибутом Avg. personnel и папку Total costs (общие затраты) с атрибутом Avg. total costs, а подгруппа Times содержит папку Orientation time (ориентировочное время) с атрибутом Avg. orientation time, папку Processing time (время процесса) с атрибутом Avg. processing time и папку Wait time (время ожидания) с атрибутом Avg. wait time.

На рис.7.1.1 для каждого типа модели (кроме функ-

ционального дерева) приведен список названий допустимых типов объектов.

На рис.7.1.2 для каждого типа модели (кроме функционального дерева) приведен список названий допустимых типов символов, т. е. возможных изображений типов объектов.

На рис.7.1.3, 7.1.4 представлены названия типов объектов и им соответствующие названия типов символов. Например, объект типа Information carrier (информационный носитель) может быть изображен одним из двадцати трех символов. В таблице 7.1.3 представлены типы детализирующих связей (Assignments). В первом столбце перечислены типы объектов, которые можно детализировать путем составления уточняющих моделей. Типы соответствующих моделей приведены во втором столбце этой таблицы. Заметим, что функцию можно детализировать, составив для нее шесть различных типов моделей. Отношения (связи) между объектами внутри модели, перечисленные в таблице 7.1.4, имеют три типа атрибута – Type, Connection role и Probability (тип, роль отношения и вероятность отношения). Только по два первых атрибута имеют связи, перечисленные в таблице 7.1.5.

### 7.2. Типы моделей и примеры их построения

Большинство типов моделей, которые допустимы в ARIS 5.0 с простым фильтром, уже рассмотрены. В предшествующем разделе приведена достаточно полная (но не исчерпывающая) информация по их объектам, связям и атрибутам. Поэтому здесь основной упор сделан на примеры построения этих моделей.

Из моделей данных простейший фильтр предоставляет только модель типа *Technical term model* (Модель технических терминов). В ней можно использовать два объекта: во-первых, *Technical term* (Технический термин), изображаемый белым прямоугольником с символами FB в правом нижнем углу, и, во-вторых, *Cluster* (Кластер), изображаемый красным прямоугольником с левой и правой сторонами из вертикальных тройных линий. Между техническими терминами допускается связь типа *has*

Таблица 7.1.1

| N   | Название атрибута      | Перевод названия     | Описание назначения атрибута  |
|---|------------------------|----------------------|---|
| <i>Атрибуты в группе (в главной папке) Attributes</i> |                        |                      |   |
| 1   | Name                   | Имя                  | Задается пользователем для всех ARIS-сущностей. Используется для выделения сущности среди других. Уникальность имени имеет важное значение. |
| 2   | Identifier             | Идентификатор        | Используется в БД, чтобы избежать проблем с синонимами или омонимами в именах сущностей.  |
| 3   | Description/Definition | Описание/Определение | Используется для краткого словесного описания (определения) объекта.  |
| <i>Атрибуты подгруппы System attributes</i>           |                        |                      |   |
| 1   | Title 1                | 1-й заголовок        | Заголовок первого документа, программы и т. д., связанных с моделью.  |
| 2   | Link 1                 | 1-я связь            | Полный путь к первому документу, программе и т. д., связанных с моделью.  |
| 3   | Parameter 1            | 1-й параметр         | Параметр, передаваемый в первый документ, программу и т. д., связанных с моделью.   |

Таблица 7.1.2

| N   | Название атрибута      | Перевод названия     | Описание назначения атрибута   |
|---|------------------------|----------------------|--|
| <i>Объектные атрибуты (атрибуты в главной группе Attributes)</i>              |                        |                      |  |
| 1   | Name                   | Имя                  | Обязательно заполняется (задается) пользователем на графическом изображении объекта. Длина имени не должна превышать 40 символов.                                  |
| 2   | Identifier             | Идентификатор        | Используется ARIS для идентификации объектов.  |
| 3   | Description/Definition | Описание/Определение | Используется для краткого словесного описания (определения) объекта.   |
| 4   | Full name              | Полное имя           | Длина полного имени не должна превышать 100 символов. Заполняется, если имя содержит более сорока символов.  |
| 5   | Remark/Example         | Заметки/Примеры      | В этом атрибуте можно привести пример использования объекта, примечание или комментарий к нему. Текст не должен превышать 3000 символов (не более одной страницы). |
| 6   | Type                   | Тип                  | Присваивается (назначается) системой.  |
| 7   | Creator                | Создатель            | Соответствует имени зарегистрированного пользователя.  |
| <i>Дополнительные атрибуты для объекта Person в главной группе Attributes</i> |                        |                      |  |
| 8   | Telephone number       | Номер телефона       | Номер телефона личности, участвующей в моделируемой системе и (или) ее БП.   |
| 9   | Fax number             | Номер факса          | Номер факса личности, участвующей в моделируемой системе и (или) ее БП.  |
| 10  | E-mail address         | E-mail адрес         | Номер электронного адреса личности, участвующей в моделируемой системе и (или) ее БП.  |
| 11  | Address                | Адрес                | Адрес личности, участвующей в моделируемой системе и (или) ее БП.  |
| <i>Атрибуты подгруппы System attributes</i>                                   |                        |                      |  |
| 12  | Title 1                | 1-й заголовок        | Заголовок первого документа, программы и т. д., связанных с объектом.  |
| 13  | Link 1                 | 1-я связь            | Полный путь к первому документу, программе и т. д., связанных с объектом.  |
| 14  | Parameter 1            | 1-й параметр         | Параметр, передаваемый в первый документ, программу и т. д., связанных с объектом.   |
| 15  | Title 2                | 2-й заголовок        | Заголовок второго документа, программы и т. д., связанных с объектом.  |
| 16  | Link 2                 | 2-я связь            | Полный путь ко второму документу, программе и т. д., связанных с объектом.   |
| 17  | Parameter 2            | 2-й параметр         | Параметр, передаваемый во второй документ, программу и т. д., связанных с объектом.  |

|                         |
|-------------------------|
| <i>Office process</i>   |
|                         |
| Event                   |
| Function                |
| Rule                    |
| Information carrier     |
| Application system type |
| Person                  |
| Person type             |
| Organizational unit     |
| Location                |
| Group                   |
| Position                |

|                             |
|-----------------------------|
| <i>Organizational chart</i> |
|                             |
| Organizational unit         |
| Person                      |
| Position                    |
| Location                    |
| Group                       |
| Person type                 |

|                         |
|-------------------------|
| <i>eEPC</i>             |
|                         |
| Function                |
| Event                   |
| Rule                    |
| Technical term          |
| Organizational unit     |
| Group                   |
| Location                |
| Position                |
| Information carrier     |
| Application system type |
| Cluster/Data model      |
| Person                  |
| Person type             |

|                                 |
|---------------------------------|
| <i>Industrial process</i>       |
|                                 |
| Function                        |
| Event                           |
| Rule                            |
| Application system type         |
| Information carrier             |
| Organizational unit             |
| Location                        |
| Person type                     |
| Person                          |
| Group                           |
| Position                        |
| Packaging material type         |
| Transport system type           |
| Operating resource type         |
| Warehouse equipment type        |
| Technical operating supply type |
| Material type                   |

|                                    |
|------------------------------------|
| <i>Function allocation diagram</i> |
|                                    |
| Function                           |
| Organizational unit                |
| Position                           |
| Person                             |
| Group                              |
| Location                           |
| Technical term                     |
| Information carrier                |
| Application system type            |
| Cluster/Data model                 |
| Person type                        |

|                                  |
|----------------------------------|
| <i>Value added chain diagram</i> |
|                                  |
| Function                         |
| Technical term                   |
| Organizational unit              |
| Cluster/Data model               |

|                              |
|------------------------------|
| <i>Technical terms model</i> |
|                              |
| Technical term               |
| Cluster/Data model           |

Рис.7.1.1 Списки используемых объектов для каждого типа модели



|                         |                       |
|-------------------------|-----------------------|
| Information carrier     | Rule                  |
| File                    | AND operator          |
| Document                | OR operator           |
| Card file               | XOR operator          |
| Expertise               |                       |
| Folder                  |                       |
| Telephone               |                       |
| Fax                     |                       |
| Bar code                | Person                |
| Magnetic tape           | Internal person       |
| Microfiche              | External person       |
| Printer                 | Person (m)            |
| Diskette                | Person (f)            |
| File cabinet            |                       |
| Notepad                 |                       |
| E-mail                  |                       |
| File bin                | Function              |
| CD-ROM                  | Process interface     |
| Hard disk               | Function              |
| Wastepaper basket       | (Manufacturing)       |
| Letter                  | Function (Office)     |
| Book                    | Value-added chain     |
| Time planner            |                       |
| Internet                |                       |
|                         |                       |
| Operating resource type | Transport system type |
| Machine                 | Airplane              |
| Robot                   | Truck                 |
|                         | Transport system      |

Рис.7.1.3 Списки типов символов для каждого объекта.  
Совпадающие названия объектов и их символов не приводятся.

|                     |                                 |
|---------------------|---------------------------------|
| Cluster             | Cluster/Data model              |
| Computer            | Application system type         |
| Packaging Material  | Packaging material type         |
| Warehouse equipment | Warehouse equipment type        |
| Tool                | Technical operating supply type |

Рис.7.1.4 Списки типов символов (слева) и соответствующих типов объектов (справа).

Таблица 7.1.3

| Тип объекта         | Тип модели                  |
|---------------------|-----------------------------|
| Function            | Value added chain diagram   |
| Function            | eEPC                        |
| Function            | Function allocation diagram |
| Function            | Function tree               |
| Function            | Office process              |
| Function            | Industrial process          |
| Group               | Organizational chart        |
| Location            | Organizational chart        |
| Organizational unit | Organizational chart        |
| Technical term      | Technical terms model       |

Таблица 7.1.4

| Тип отношения   | Перевод                 |
|-----------------|-------------------------|
| activates       | активизирует            |
| creates         | порождает               |
| is evaluated by | оценивается с помощью   |
| leads to        | порождает событие через |
| links           | связывает               |

*relation with* (“имеет связь с”), от технического термина к кластеру проводится связь типа *depicts* (“изображает”), от кластера к термину связь провести нельзя, и между кластерами связь не допускается.

Кластер представляет собой логическое представление набора сущностей, которые описывают сложное понятие. Кластер может объединять в себе понятия, которые классифицируют описываемую сущность. Он может использоваться для отражения логического взгляда на набор описываемых терминов (понятий) и т. д. На рис. 7.2.1.–7.2.3 приведены примеры моделей терминов, которые описывают их взаимосвязь и отражают рассматриваемую в этой книге предметную область – ARIS-моделирование.

При построении диаграммы типа *Organizational chart* можно использовать объекты и типы связи между ними, которые изображены на рис. 7.2.4. При этом рекомендуется придерживаться следующей методики. Сначала построить иерархическое дерево, отражающее либо должностную иерархию на предприятии (диаграмму подчиненности субъектов ответственности предприятия на основе объектов *Person type* или *Position*), либо организационную иерархию (диаграмму взаимосвязей организационных структур предприятия на основе объекта *Organizational unit*), либо иерархию расположения подразделений и рабочих мест предприятия (диаграмму размещения подразделений предприятия на основе объекта *Location*). Затем с каждым из объектов получившегося иерархичес-

Таблица 7.1.5

| Тип отношения                | Перевод                                  |
|------------------------------|--|
| accesses                     | имеет доступ к                           |
| belongs to                   | принадлежит                              |
| can access                   | может иметь доступ к                     |
| can be user                  | может использоваться                     |
| can subsume                  | может являться обобщающей категорией для |
| can support                  | может поддерживать                       |
| can use                      | может использовать                       |
| cooperates with              | взаимодействует с                        |
| creates output to            | создает на выходе                        |
| depicts                      | описывает                                |
| executes                     | выполняет                                |
| has member                   | имеет в своем составе                    |
| has output of                | имеет на выходе                          |
| has relation with            | имеет отношение к                        |
| has state                    | отображает                               |
| is assigned to               | связан с                                 |
| is composed of               | состоит из                               |
| is executed at               | выполняется на территории                |
| is input for                 | является входом для                      |
| is located at                | располагается                            |
| is operating resource of     | является ресурсом для                    |
| is Organization Manager for  | является организационным управляющим     |
| is partly consumed           | частично потребляется                    |
| is predecessor of            | предшествует                             |
| is process-oriented superior | подчиняется по процессу                  |
| is superior                  | имеет в подчинении                       |
| lies on                      | отображается                             |
| occupies                     | занимает                                 |
| provides input for           | предоставляет входные данные для         |
| requires                     | требует                                  |
| subsumes                     | содержит, объединяет                     |

кого дерева связать недостающие для построения организационной структуры объекты и при необходимости связать эти объекты друг с другом. В результате из трех получившихся диаграмм можно выбрать лучшую на ваш взгляд или с точки зрения заказчика модель.

В высших учебных заведениях часто бывает так, что кафедры и даже факультеты создаются под конкретного человека, т. е. с условием, что их возглавит конкретная личность. Поэтому создание организационной диаграммы можно начать с построения дерева субъектов, отражающего их важность с точки зрения вклада в развитие организации (диаграмма значимости). Но такой подход можно применить только для тех организаций, в которых орга-



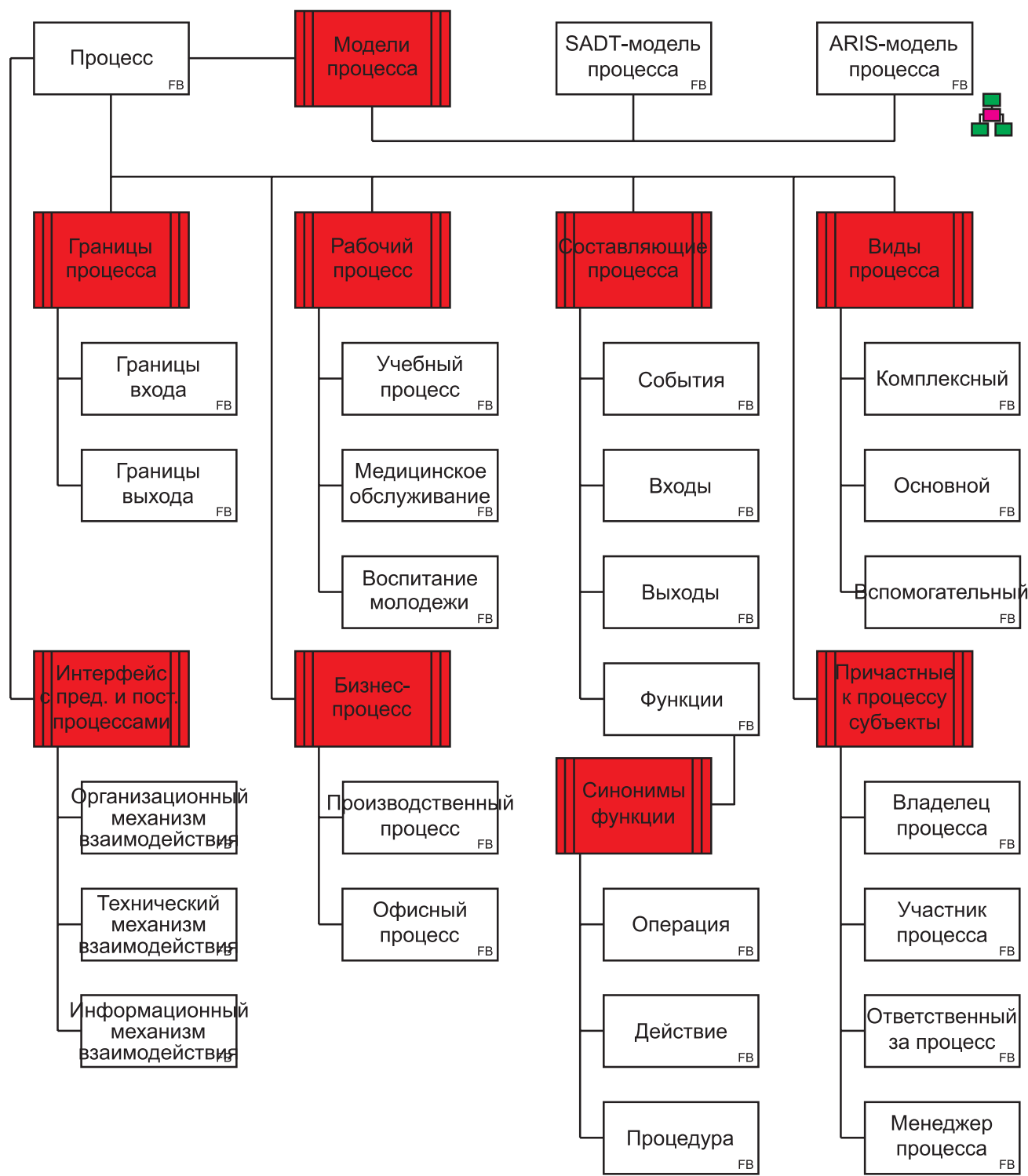


Рис.7.2.1 Модель технических терминов, связанных с понятием “Процесс”.

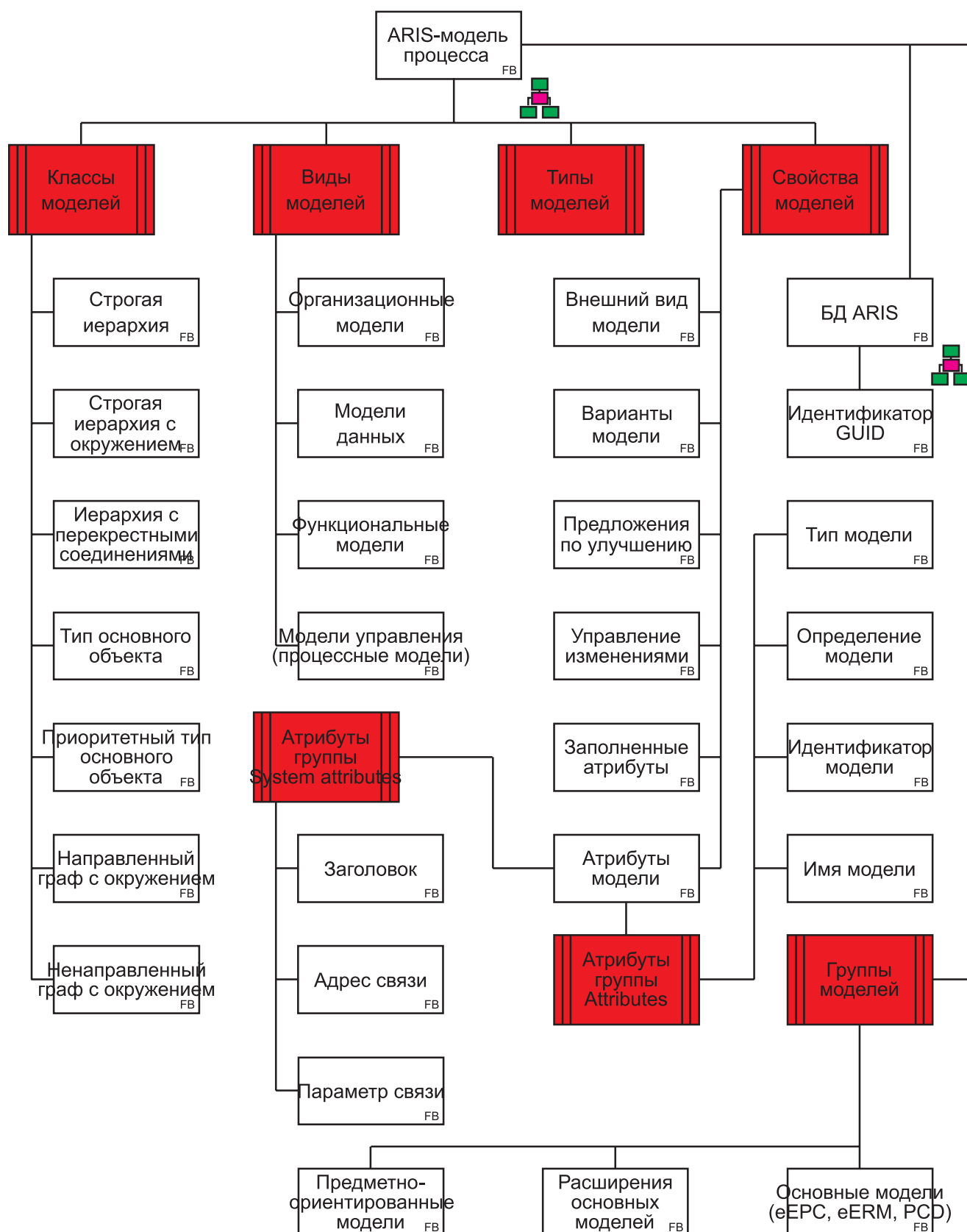


Рис.7.2.2 Модель технических терминов, связанных с понятием "ARIS-модель процесса".

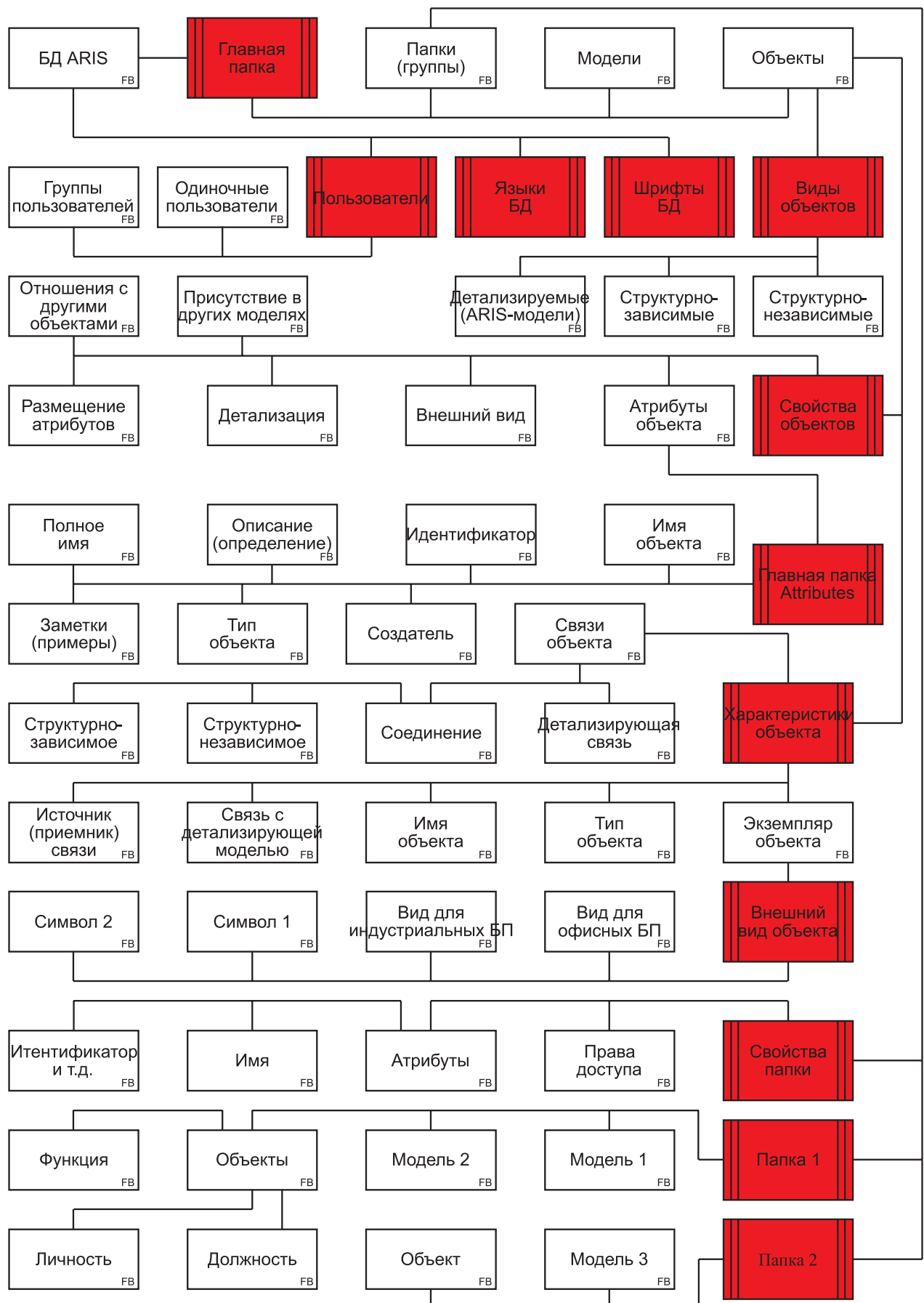


Рис.7.2.3 Модель технических терминов, связанных с понятием “БД ARIS”.

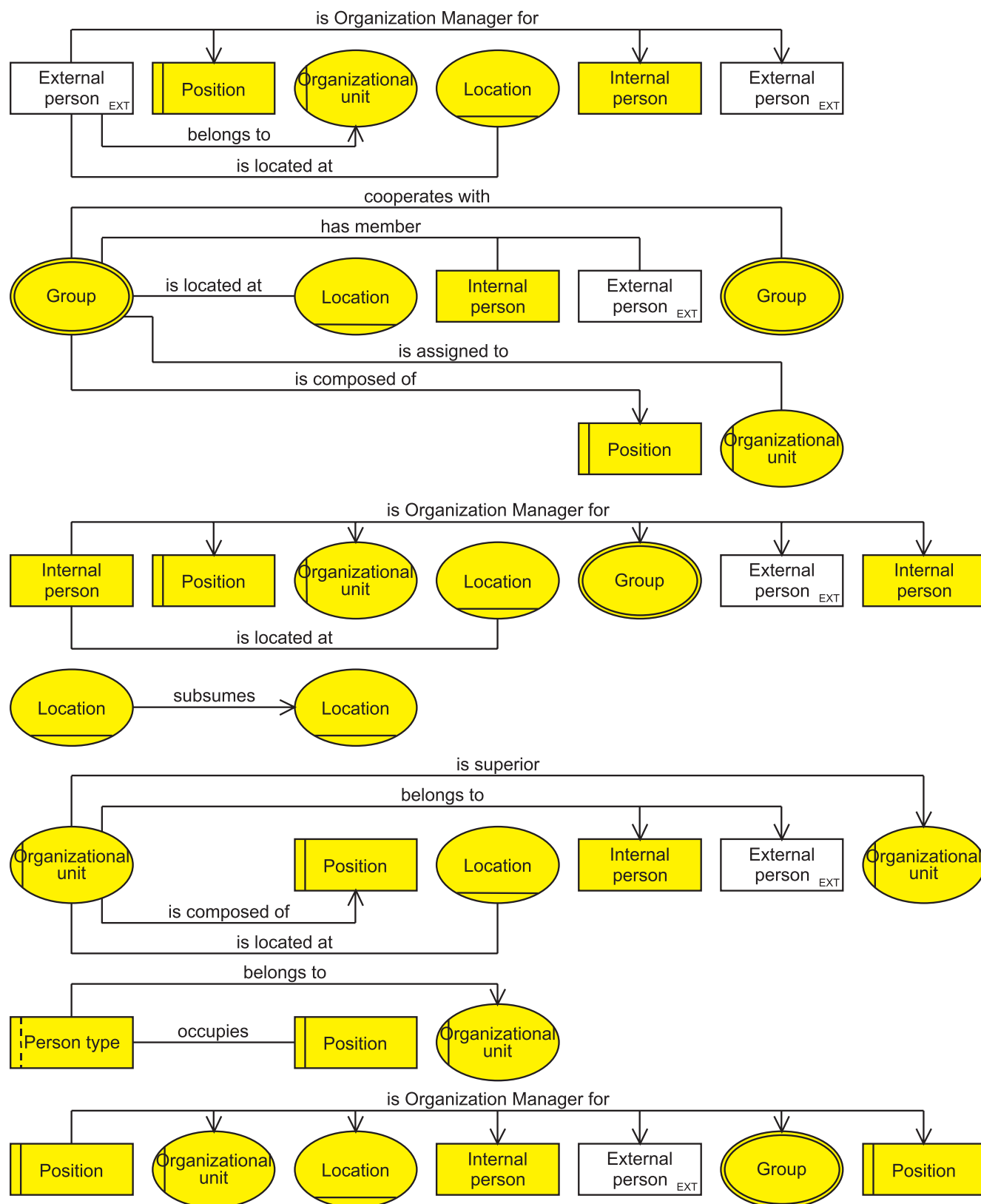


Рис.7.2.4 Объекты и связи в моделях типа Organization chart.

низационное устройство отражает сложившиеся субъективные традиции (семейственности или династий). На рис. 7.2.5-7.2.10 приведены примеры описания организационной структуры ЭТФ тремя способами. Причем на рис. 7.2.5-7.2.7 представлены их возможные базовые иерархические деревья, а на рис. 7.2.8-7.2.10 эти деревья дополнены остальными объектами.

Построение моделей БП рекомендуется начинать с построения процессно-ориентированного дерева функций. Это единственный тип модели, который допустим в функциональном виде с простым фильтром моделирования. Между экземплярами единственного типа объекта “Функция” устанавливается единственный тип связи – *is process-oriented superior* (“является процессно-ориентированной вышестоящей”). Все функции, к которым проведена эта связь, являются участниками одного БП. Название этого БП является названием функции, от которой исходят эти связи. В качестве примера на рис. 7.2.11. изображено процессно-ориентированное дерево для БП, протекающего по следующему сценарию.

Между предприятием и поставщиком материалов существует договор, по которому поставщик доставляет предприятию заказанные материалы по заранее оговоренным условиям отгрузки, доставки, оплаты и т. д. По этим условиям поставщик материалов через транспортную организацию (ТрО) доставляет заказанные предприятием материалы. Оплата за доставку осуществляется после их получения предприятием. Поставщик, отгружая транспортной организации материалы, выписывает два экземпляра счета-фактуры (СФ), один из которых оставляет у себя для отчетности, а другой передает транспортной организации для перевозки материалов и вручения предприятию вместе с заказанными материалами. ТрО при отгрузке выписывает путевой лист, поставщик – три экземпляра товарно-транспортной накладной (ТТН), которые отправляются вместе с заказанными материалами предприятию. Предприятие, принимая у транспортной организации материалы, делает отметку в двух экземплярах товарно-транспортной накладной, один из которых возвращается в транспортную организацию для отчетности, а третий оставляет у себя. На основании полученного счета-фактуры предприятие перечисляет деньги за заказанные материалы и их доставку поставщику, который после получения денежных средств рассчитывается с транспортной организацией.

На рис. 7.2.11 не отражена логика и последовательность выполнения функций в цепи рассмотренного БП. Эта информация отражается в eEPC-модели БП.

Перечень объектов и их связей, допустимых в eEPC-

моделях, изображены на рис. 7.2.12.

На рис. 7.2.13 на основе процессно-ориентированного дерева и описанного сценария построена eEPC-модель БП “Доставка материалов”.

Выполнение сложной функции из eEPC-диаграммы может быть детализировано в другой eEPC-модели. Для сложно выполняемых функций необходимо строить три диаграммы. Во-первых, процессно-ориентированное дерево, где перечислены все операции (подфункции), на которые разбивается выполнение сложной функции. Во-вторых, eEPC-диаграмму, в которой все подфункции увязаны в нужных последовательностях, а точки разветвления и соединения этих последовательностей описаны логическими операторами. В-третьих, диаграммы окружения функций. Остановимся подробнее на описании назначения и примерах построения диаграммы окружения функции.

Диаграммы окружения функций используются для уменьшения сложности eEPC-модели. События, входные и выходные данные, исполнители, организационные единицы и т. д., сосредоточенные вокруг функции в eEPC-модели, загромождают ее функциональный поток. Для уменьшения этой громоздкости и для лучшего восприятия последовательности и логики выполнения функций можно лишние объекты перенести в другие диаграммы. Диаграмма окружения функции предназначена как раз для того, чтобы описать все объекты, которые окружают функцию, т. е. исполнителей, входные и выходные потоки информации, документы и т. д. Этот тип модели целесообразно применять для статической детализации функций в модели eEPC. В диаграмме окружения функции отражаются дополнительные связи и отношения, детализирующие эту функцию на уровне данных.

В диаграммах окружения функции удобно описывать преобразование входных данных в выходные. На них связи со стрелками показывают, представляет ли собой информационный объект элемент входных данных, выходных данных или является входным и выходным информационным объектом. Кроме того, можно ввести подробные указания, которые содержат информацию о том, что функция создает или удаляет информационный объект. В зависимости от степени детализации информационный объект может быть показан кластером данных, или документом, или только техническим термином.

Как правило, диаграмма окружения функции предназначена для представления всех объектов, которые имеют отношение к выполнению функции в цепи БП. Поэтому при моделировании БП в диаграмме eEPC достаточно использовать события и функции и затем назначить каждой функции диаграмму ее окружения, где показать все остальные объекты, имеющие отношение к этой функции

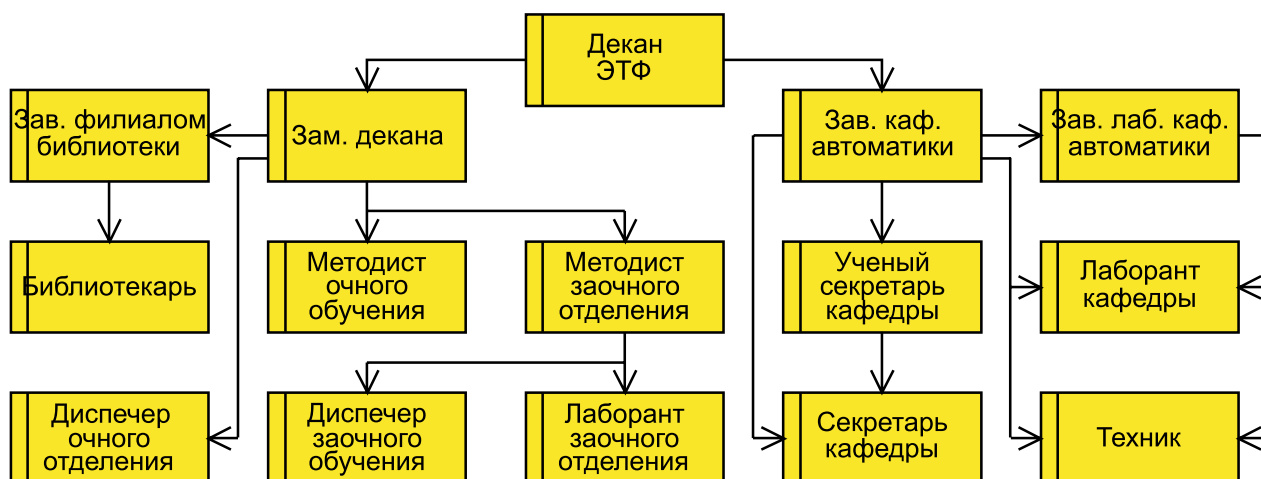


Рис.7.2.5 Фрагмент диаграммы должностной подчиненности на ЭТФ

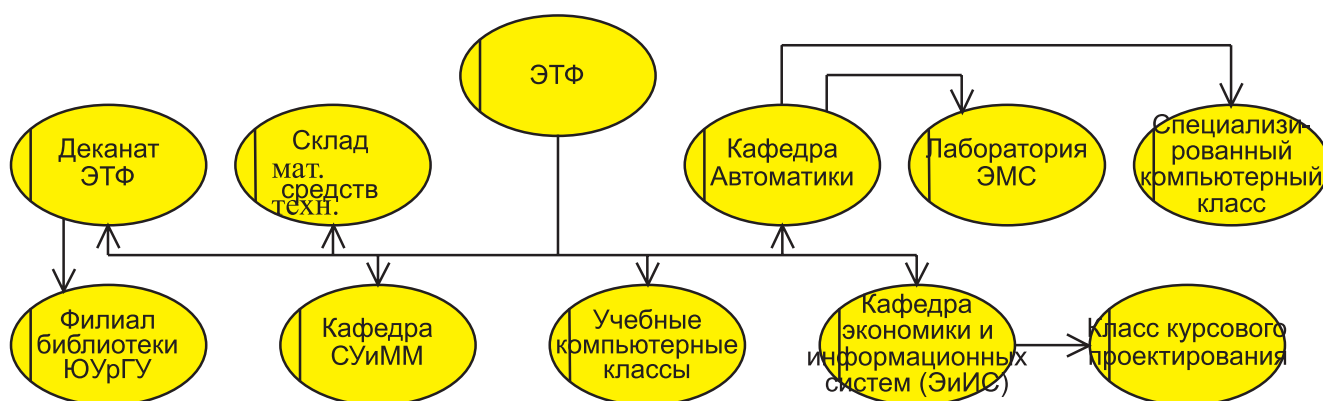


Рис.7.2.6 Фрагмент диаграммы взаимосвязей организационных единиц ЭТФ

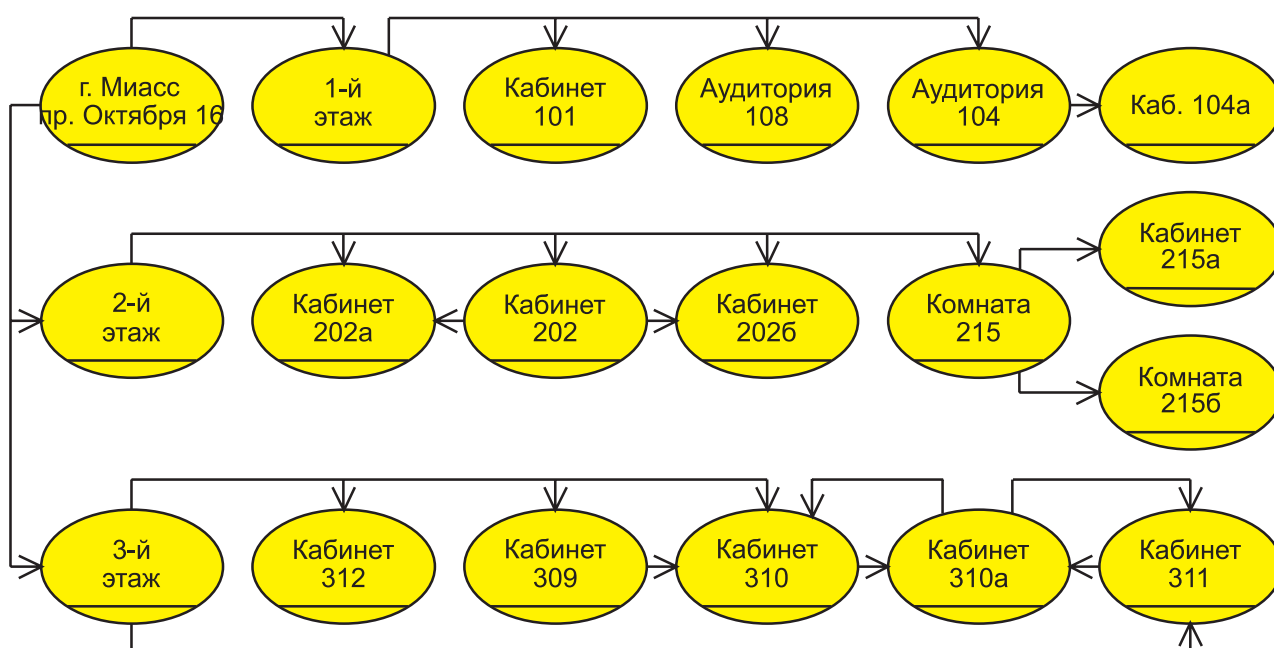


Рис.7.2.7 Фрагмент диаграммы размещения ЭТФ



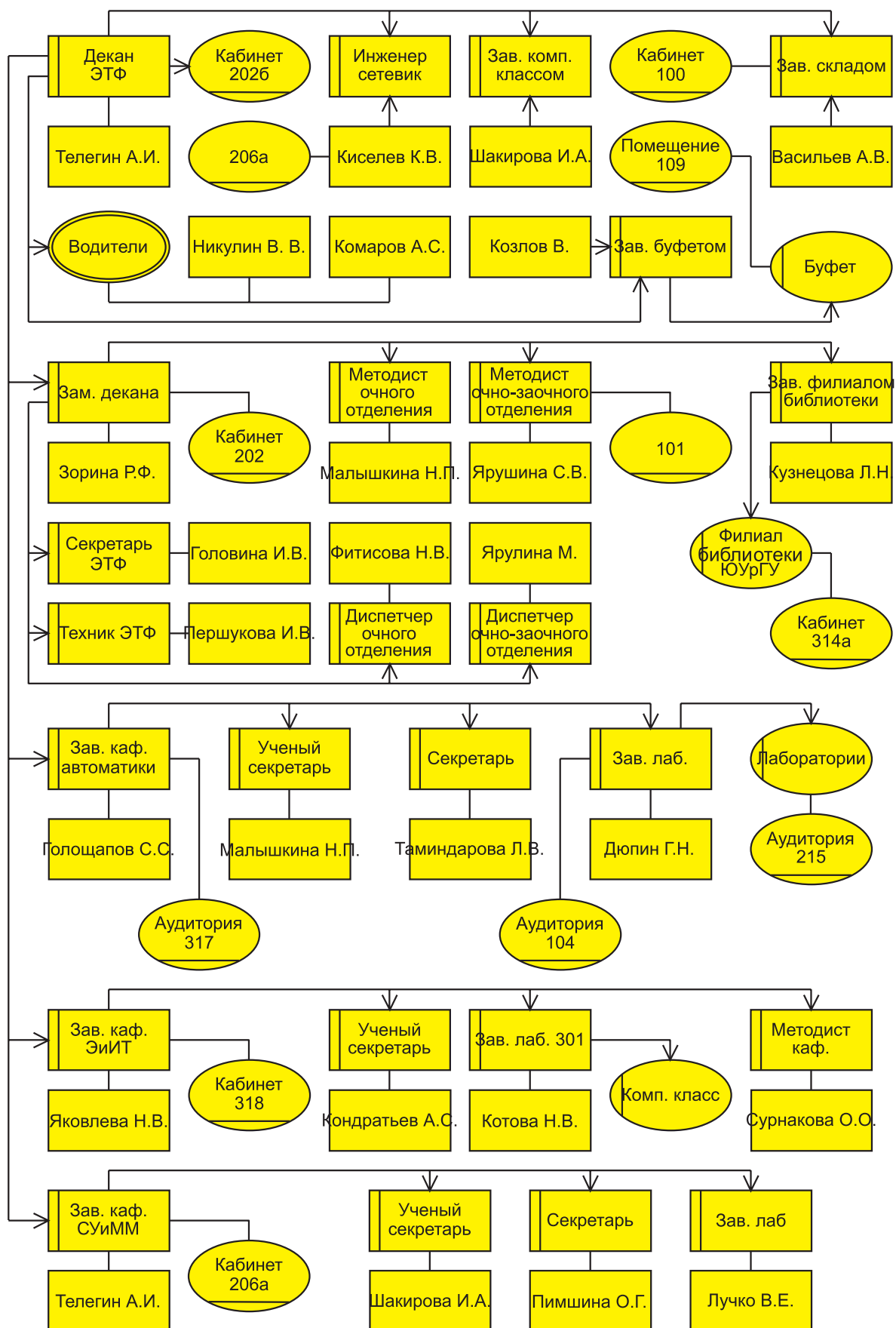


Рис.7.2.8 Фрагмент организационной диаграммы ЭТФ, построенной на основе диаграммы должностной подчиненности

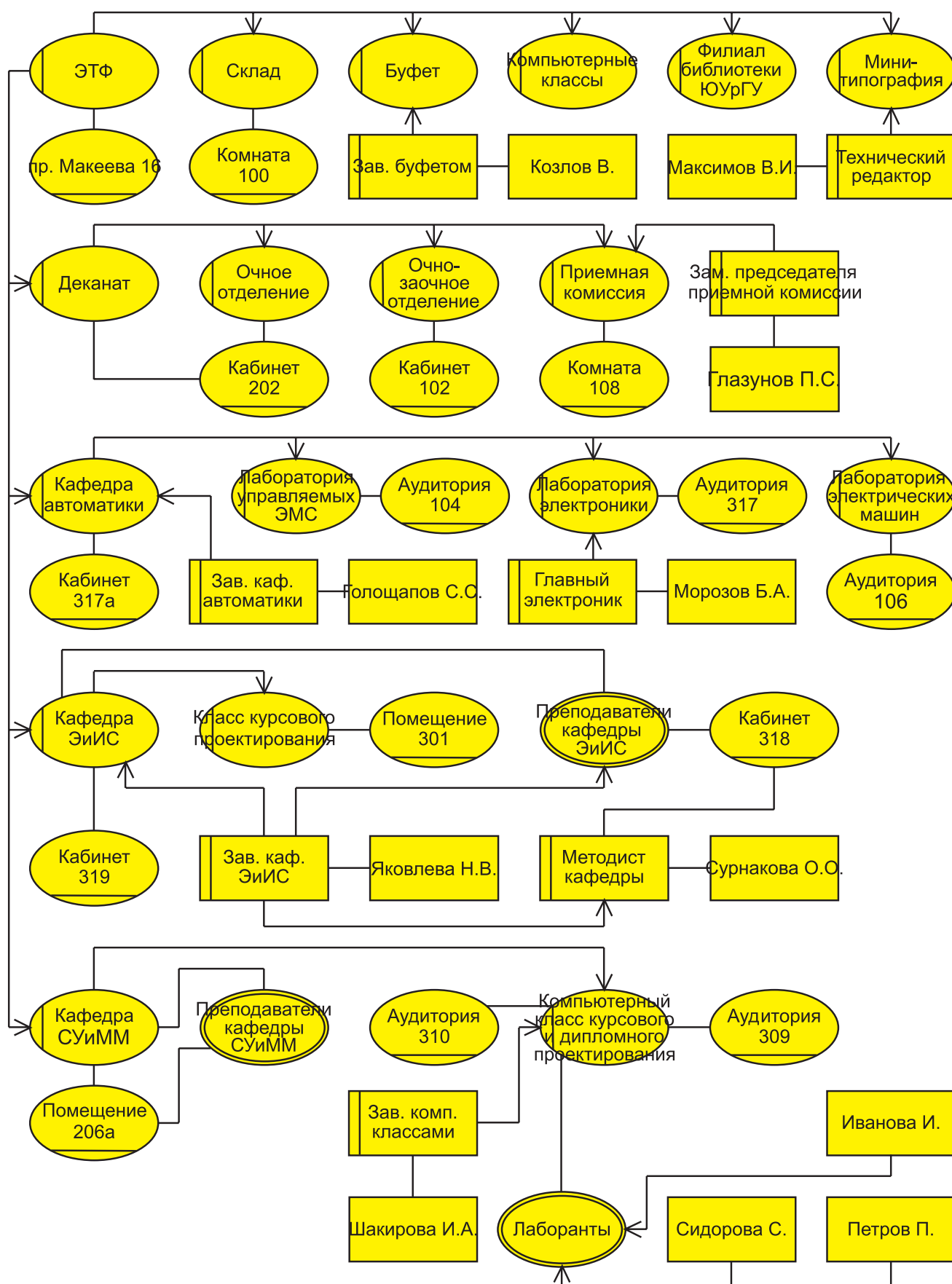


Рис.7.2.9 Фрагмент организационной диаграммы ЭТФ, построенной на основе диаграммы организационных единиц

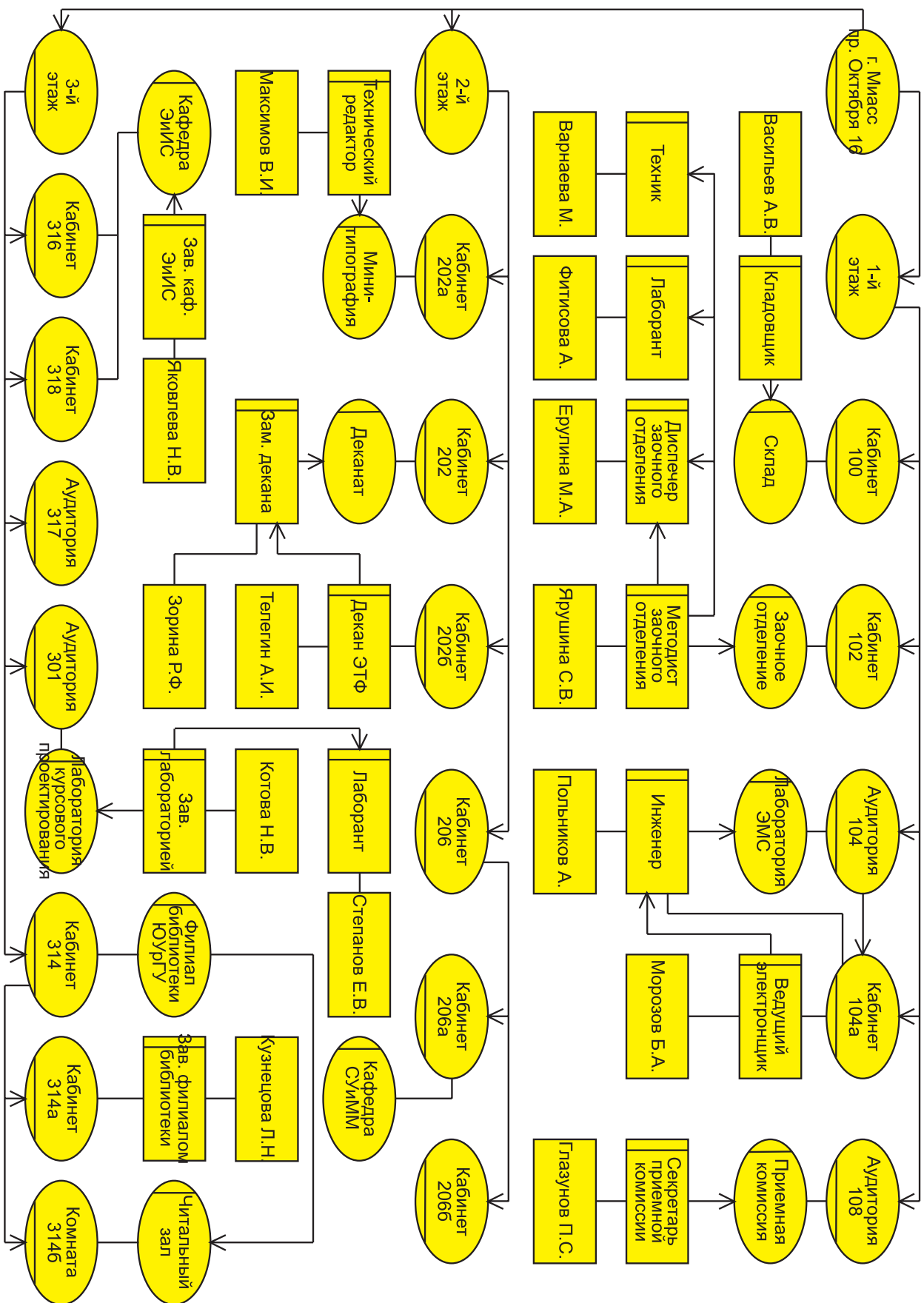


Рис.7.2.10 Фрагмент организационной диаграммы ЭТФ, построенной на основе диаграммы размещения

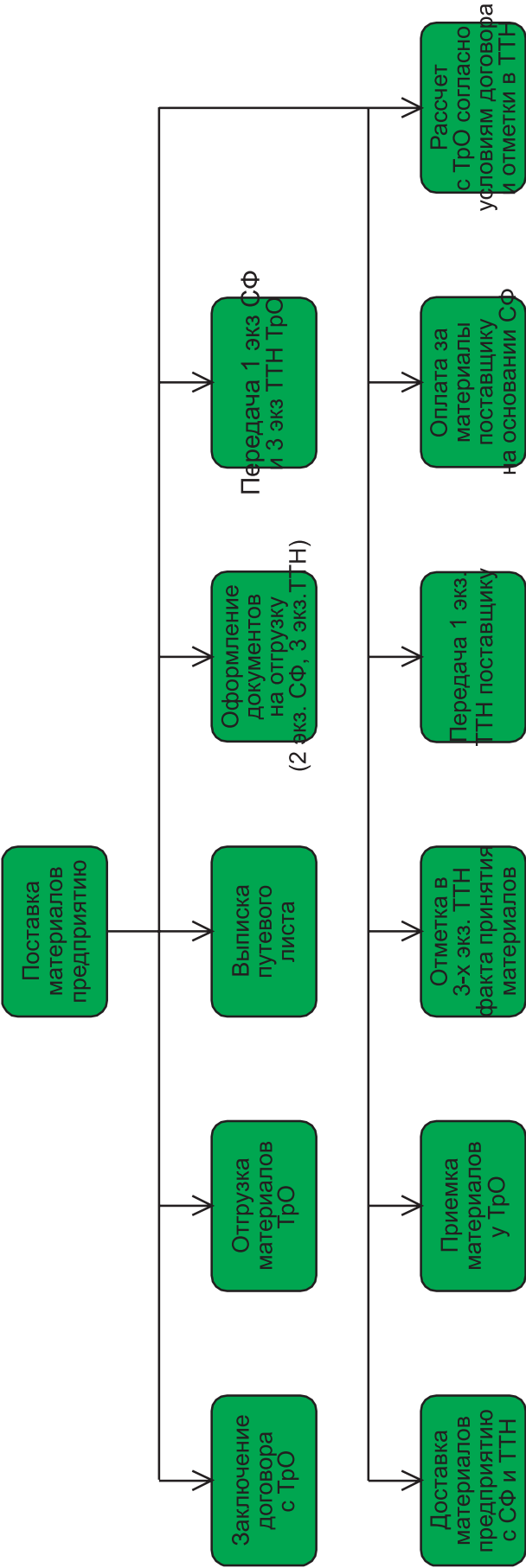


Рис.7.2.11 Процессно-ориентированное дерево

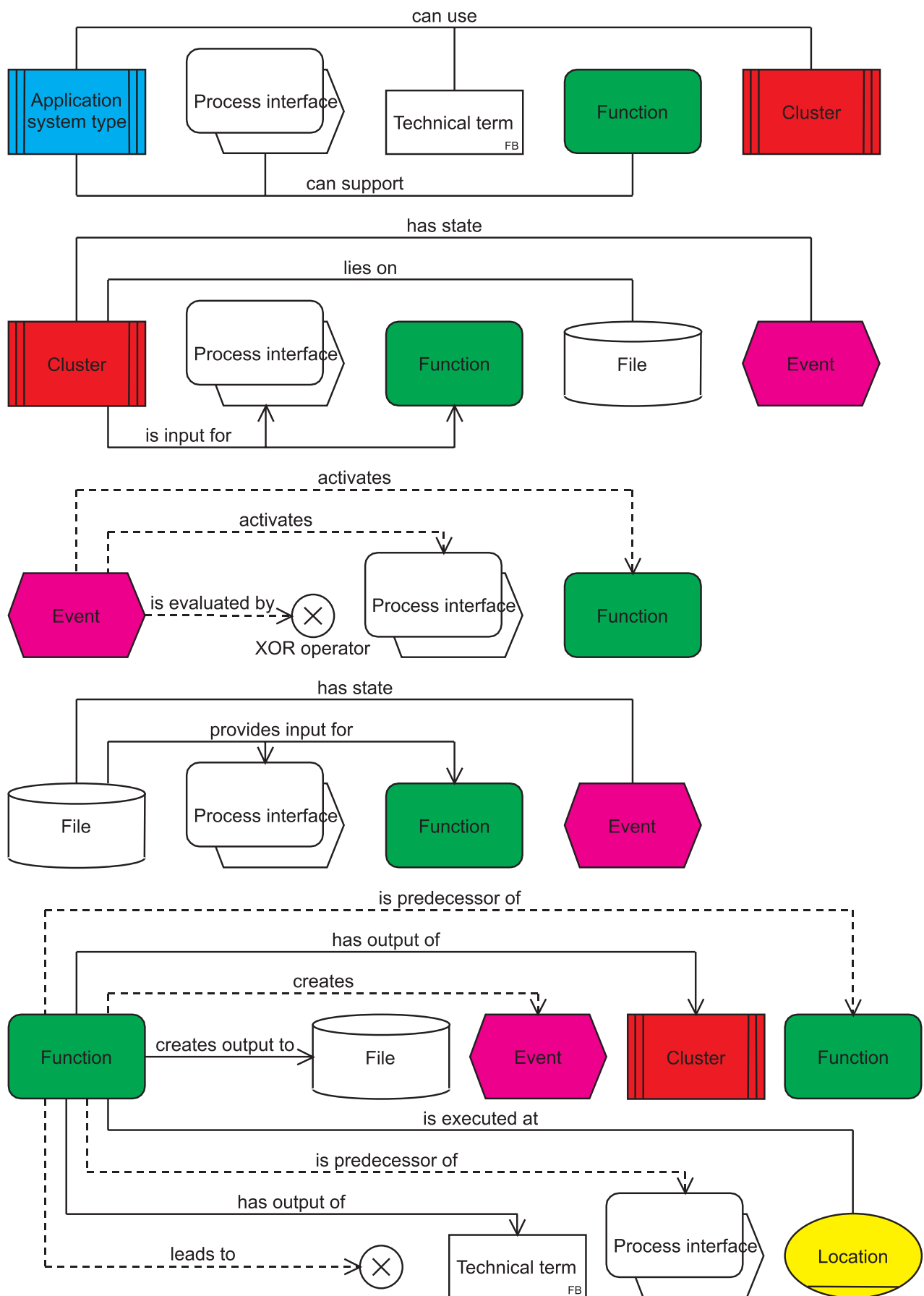


Рис.7.2.12 Объекты и связи в eEPC-моделях (начало)

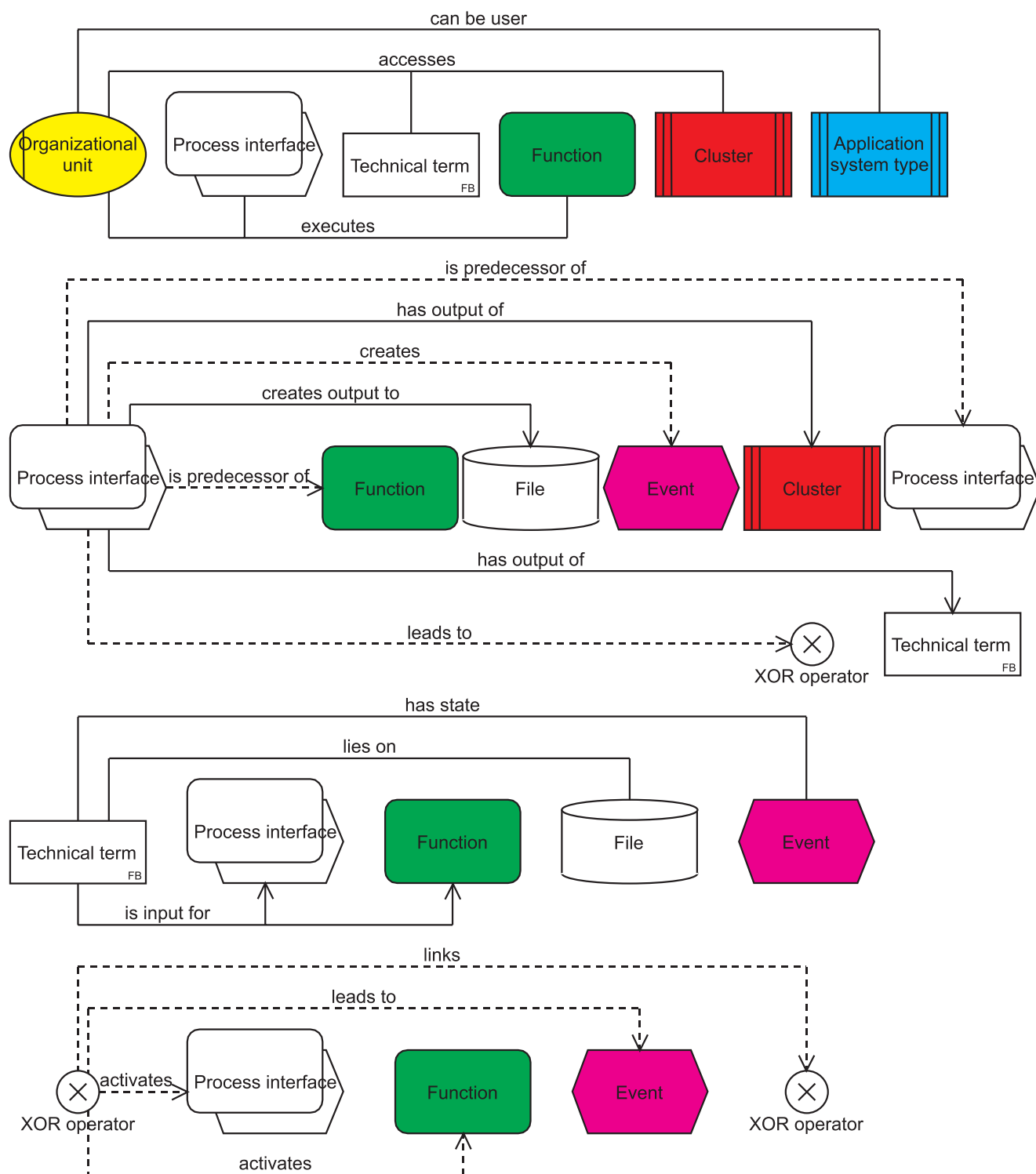


Рис.7.2.12 Объекты и связи в eEPC-моделях (окончание)



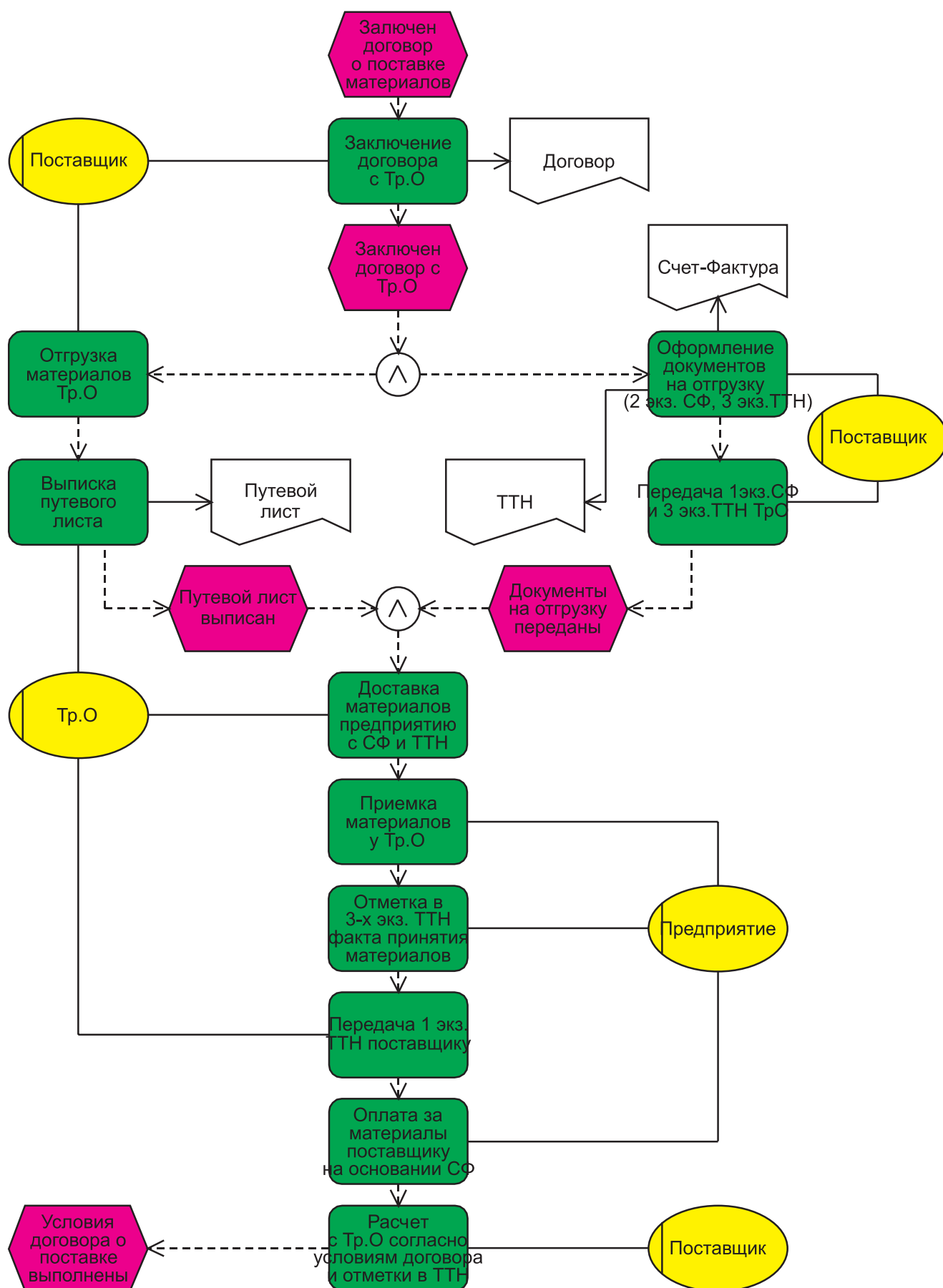


Рис.7.2.13 eEPC-диаграмма поставки материалов предприятию

в рамках моделируемого БП. В результате получим ясное представление о БП.

На рис.7.2.14 представлены символы объектов, на множестве которых строятся eEPC-модели. Если из этого перечня убрать два верхних символа – Event и Process interface – и три логических оператора – “и”, “или” и “исключающее или”, – то получим набор символов объектов, на множестве которых строится диаграмма окружения функций. На рис.7.2.15 представлен общий вид такой диаграммы, а на рис.7.2.16 отдельно изображены объекты-источники связей и для каждого из них допустимые объекты-приемники связей. На рис.7.2.14 по пять последних объектов в столбцах, а на рис.7.2.15 в двух последних строках изображены носители информации. Эти десять символов объектов относятся к типу *Information carrier* (“Носитель информации”), и на многих рисунках этого раздела их представляет объект *File* (“Файл”).

eEPC-диаграмма может детализировать одну из функций диаграммы типа *Value-added chain diagram* (Цепочка добавленной стоимости (качества)). Но не наоборот, т. е. цепочка добавленной стоимости (качества) не может детализировать одну из функций eEPC-диаграммы.

Диаграммы типа *Value-added chain diagram* применяются для описания БП сверху вниз, где сначала описываются суперпроцессы, а затем функции этих процессов последовательно детализируются. На рис.7.2.17 представлены допустимые в *Value-added chain diagram* типы объектов и связи между ними. На рис.7.2.18 приведен пример модели типа *Value-added chain diagram*.

В заключение рассмотрим особенности построения офисных и промышленных БП. Набор объектов в каждом из них представлен на рис.7.2.19, 7.2.20. На рис.7.2.21, 7.2.22 описаны допустимые типы связей между объектами в этих моделях. Следует заметить, что модель типа eEPC можно представить в виде модели типа *Office process* и *Industrial process*. Все зависит от того, к чему ближе БП – к офисному или промышленному. Если в БП работают только с информацией в офисе, то его удобно описать как БП типа *Office process*. Если в БП производят продукцию с использованием технологического оборудования, материалов, оснастки и т. д., то такой БП удобно описать как БП типа *Industrial process*.

Здесь мы рассмотрели все типы моделей в отрыве друг от друга. В следующей главе рассматриваются примеры, где все эти модели участвуют в описании одной системы и ее процессов.

### 7.3. Формирование ARIS-отчетов по моделям

ARIS-отчеты (далее отчеты) могут генерироваться для БД (Databases), групп (Groups), моделей (Models), объектов (Objects) и других ARIS-сущностей. Они представляют собой гибкое и эффективное средство сбора данных об ARIS-сущностях и вывода их на печать. Отчеты предоставляют пользователю возможность вывести необходимые сведения о моделях, объектах, группах и т. д. в требуемом виде. Для каждой из своих сущностей ARIS предлагает сценарии отчета определенного типа. Сценарий представляет собой ARIS-программу, которая дает возможность создания отчета по определенному шаблону с выводом требуемых данных согласно выбранным параметрам. Тип сценария учитывает особенности конкретной сущности и предоставляет пользователю определенную информацию о ее свойствах и атрибутах. В таблице 7.3.1 приведены некоторые типы сценариев и расширения файлов, их содержащих.

Каждый отчет состоит из общей и специальной части. В общей части содержится информация о дате и времени генерирования отчета, сервере, содержащем БД, имени БД, используемом фильтре, пользователе, который генерировал отчет. Содержание специальной части сценария зависит от его типа и от тех параметров, которые выбираются пользователем в процессе создания отчета.

В основе создания отчета лежит тип сценария. По файловому расширению типа сценария можно определить, для какой сущности предназначен данный отчет. Отчет формируется либо в результате выполнения ранее созданного макроса, либо в результате выполнения мастера отчетов. Рассмотрим создание отчета с помощью макроса.

Макросы используются в случаях, когда необходимо многократно выполнять какую-то задачу. *Макрос* – это набор команд, сгруппированных вместе для упрощения часто выполняемой работы. Вместо того, чтобы многократно вручную делать отнимающие много времени повторяющиеся действия, можно создать один макрос, который будет выполнять эти действия, и затем запускать его по мере необходимости. Макросы используются для следующих целей: ускорения часто выполняемых операций, объединения сложных команд, упрощения доступа к параметрам в ДО, автоматизации обработки сложных последовательных действий в задачах.

Для создания макроса необходимо выбрать пункт меню *Evaluate / Manage Macros*, после чего откроется

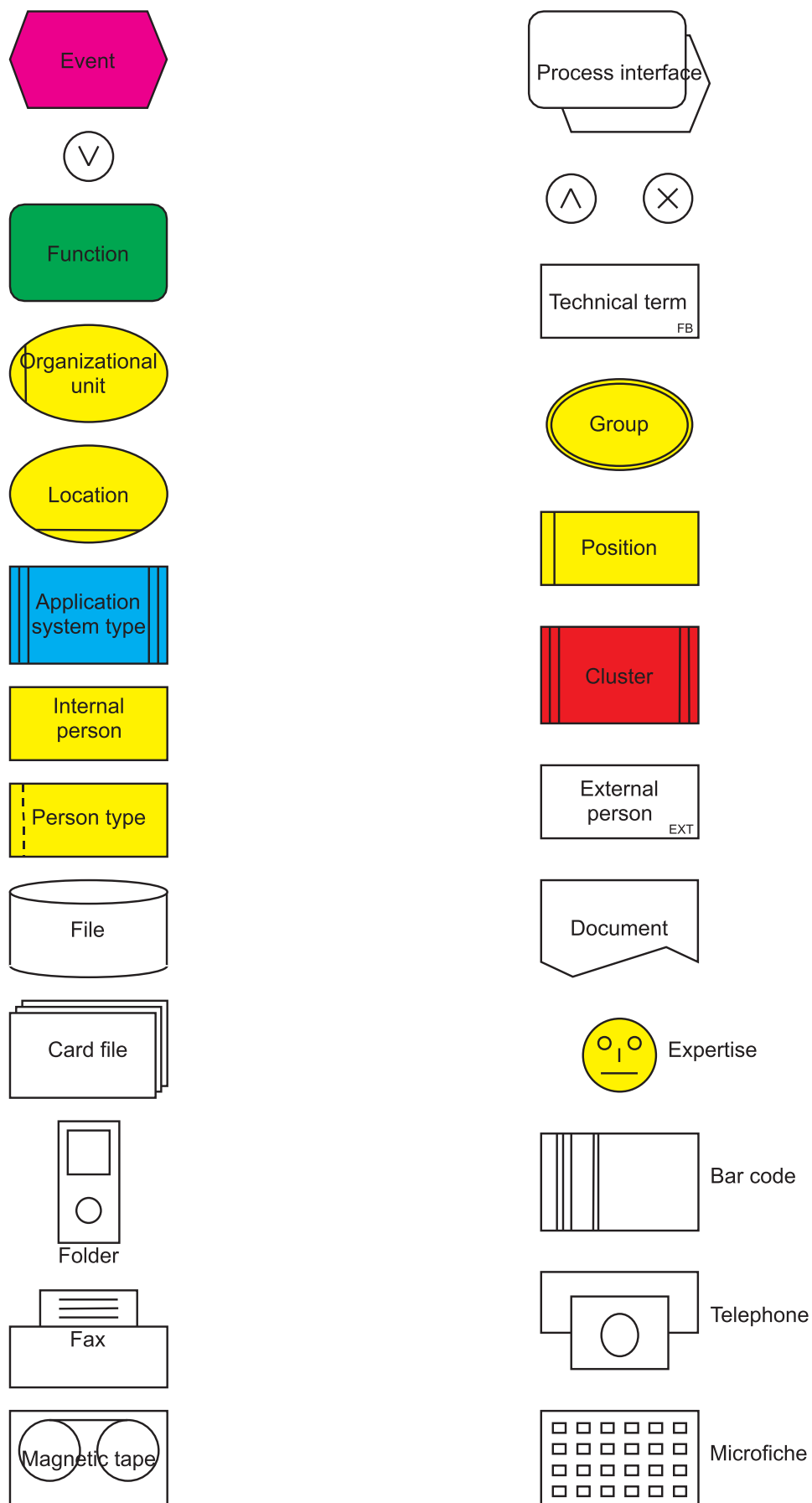


Рис.7.2.14 Типы символов объектов модели типа eEPC

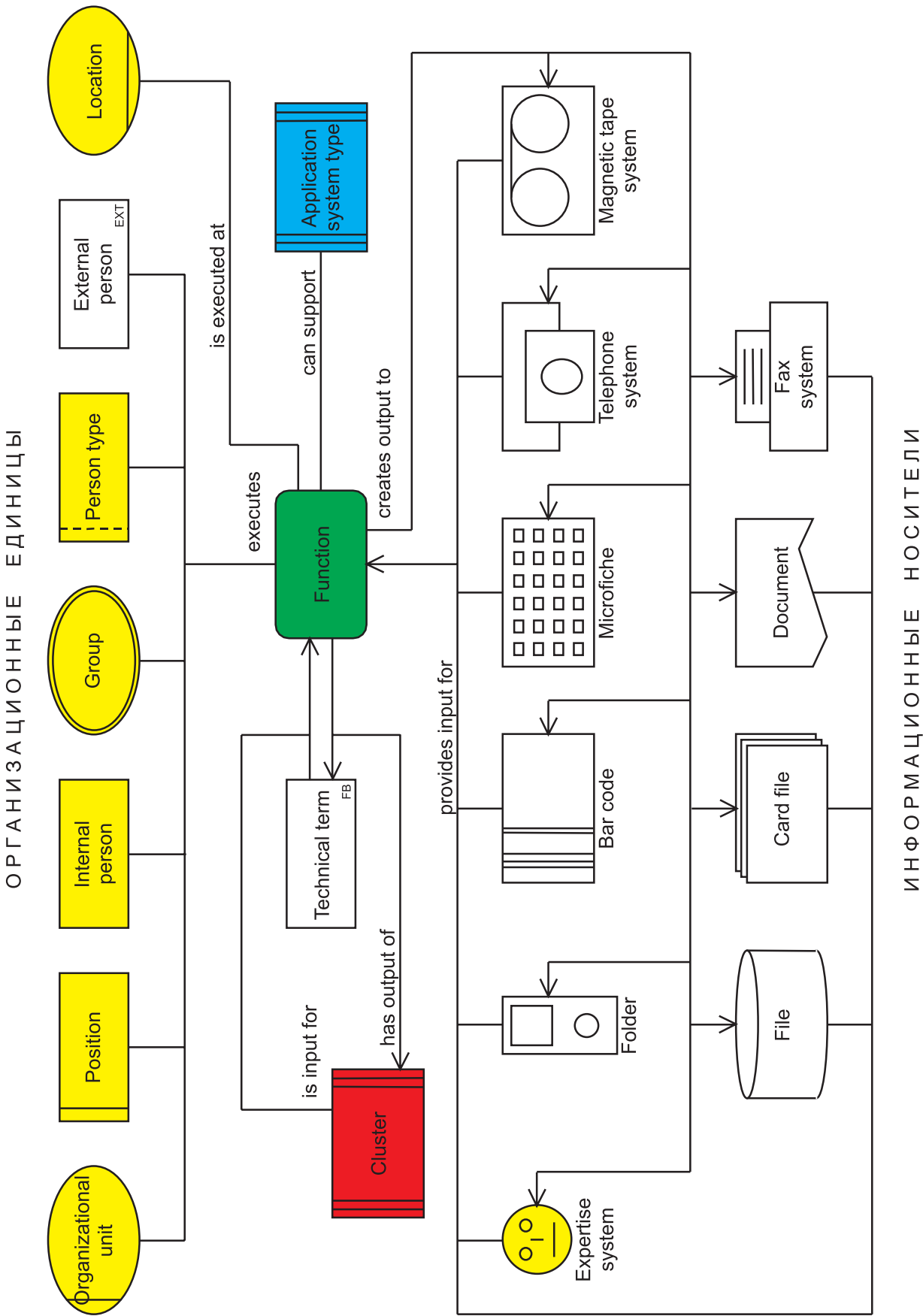


Рис.7.2.15 Общий вид диаграммы окружения функции

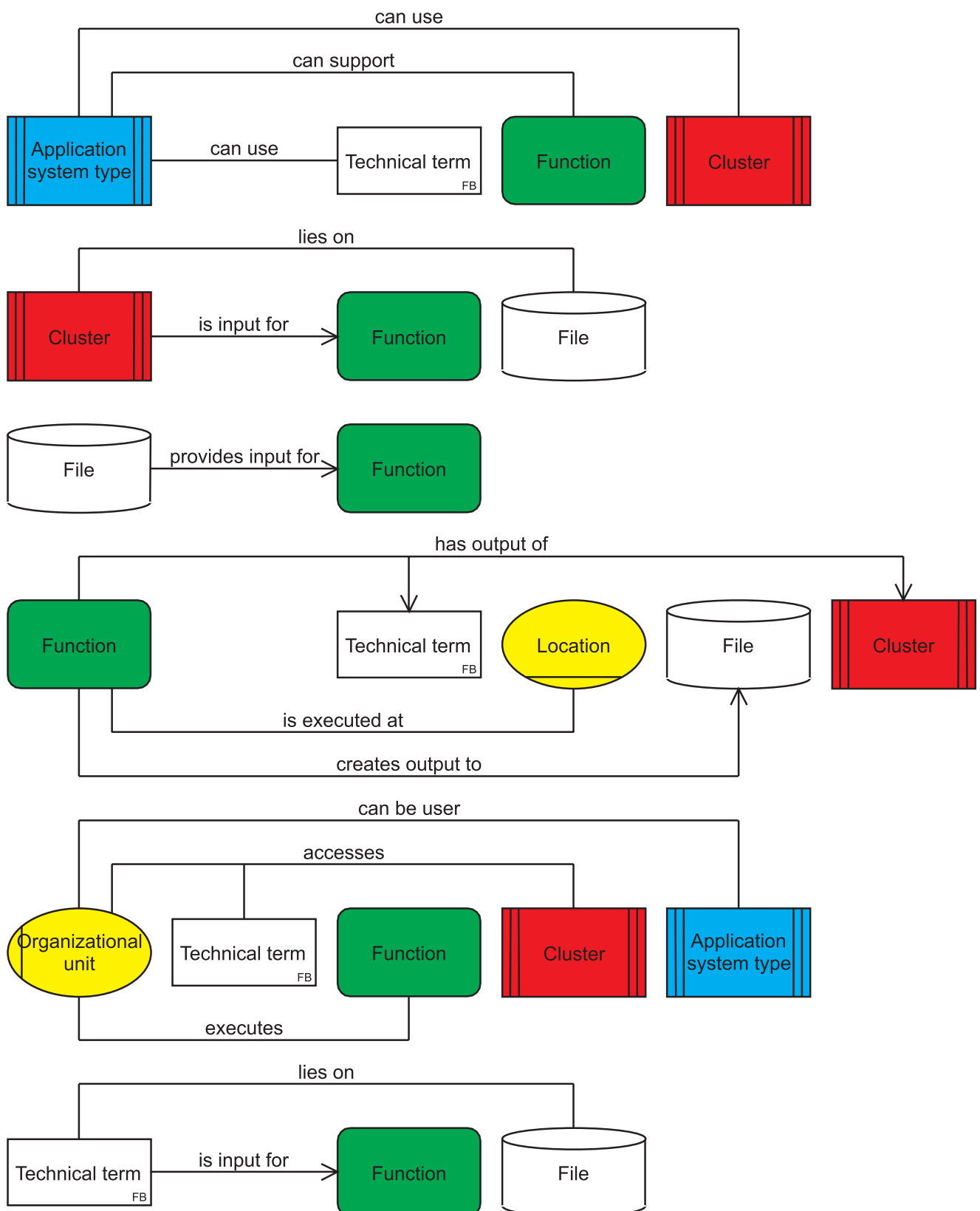


Рис.7.2.16 Перечень объектов и их связей в моделях типа Functional allocation diagram

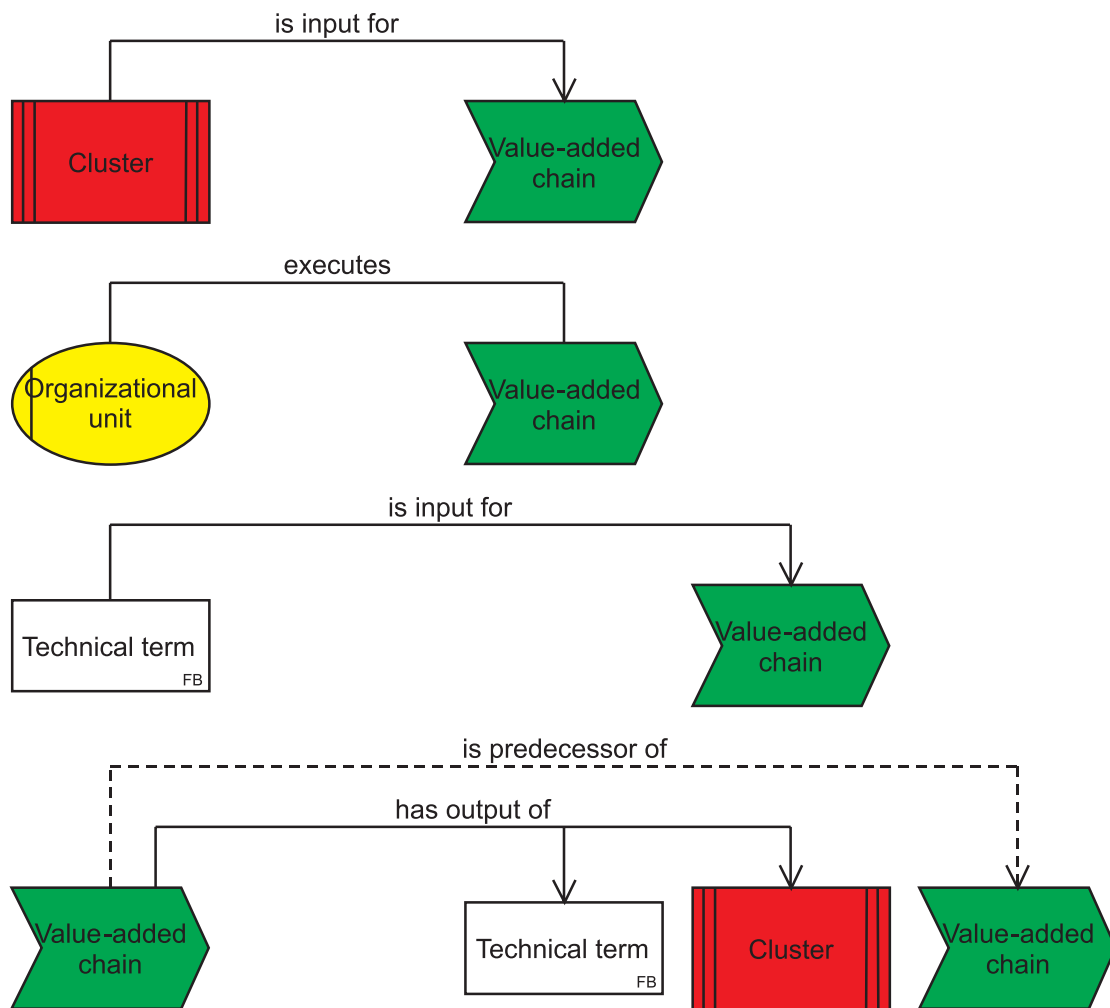


Рис.7.2.17 Объекты и связи в моделях типа Value-added chain diagram

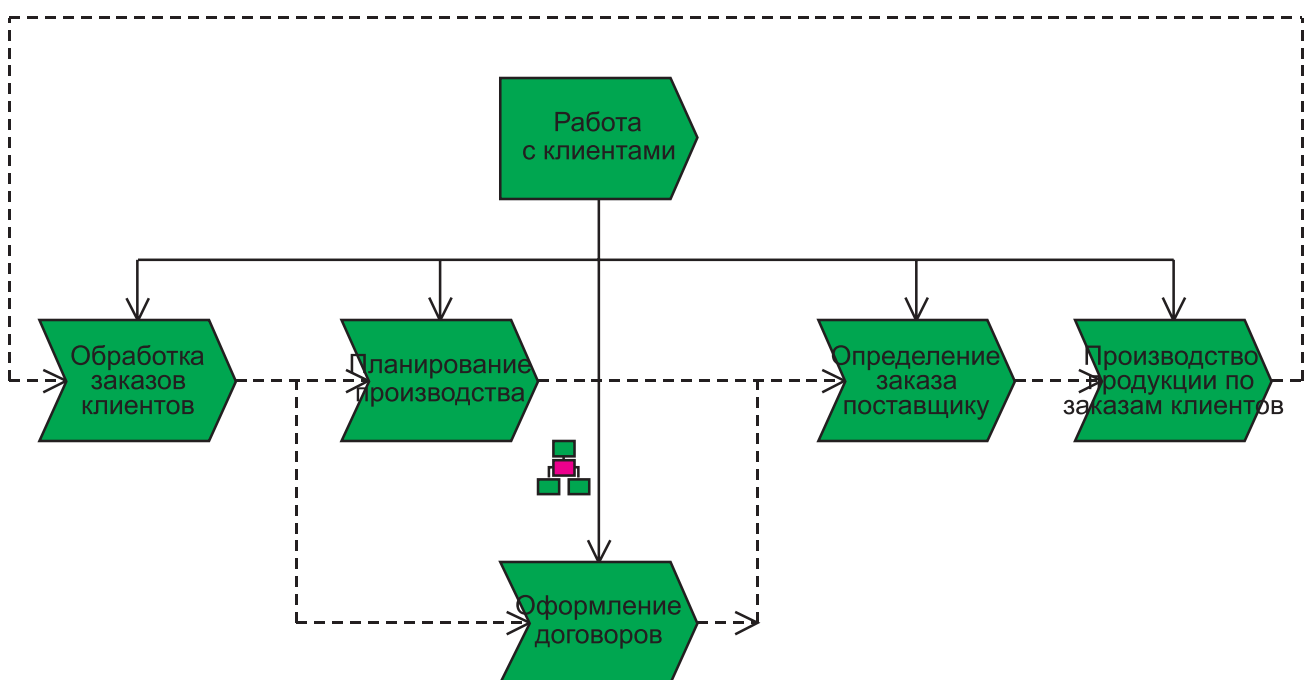


Рис.7.2.18 Пример цепочки добавленного качества





Рис.7.2.19 Список объектов модели типа Office process



Рис.7.2.20 Перечень символов (изображений) объектов, используемых в моделях типа Industrial process

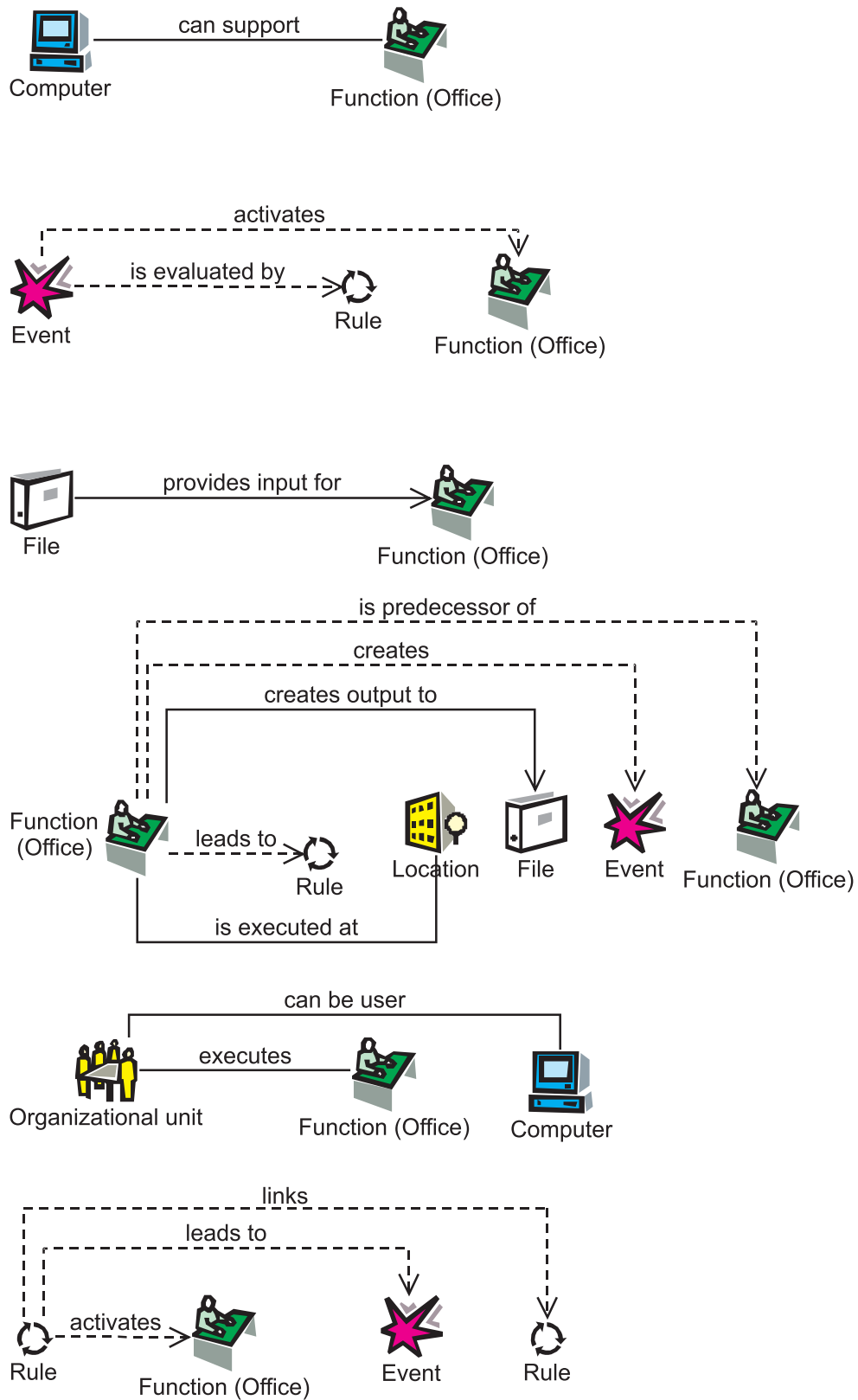


Рис.7.2.21 Объекты и их связи в моделях типа Office process

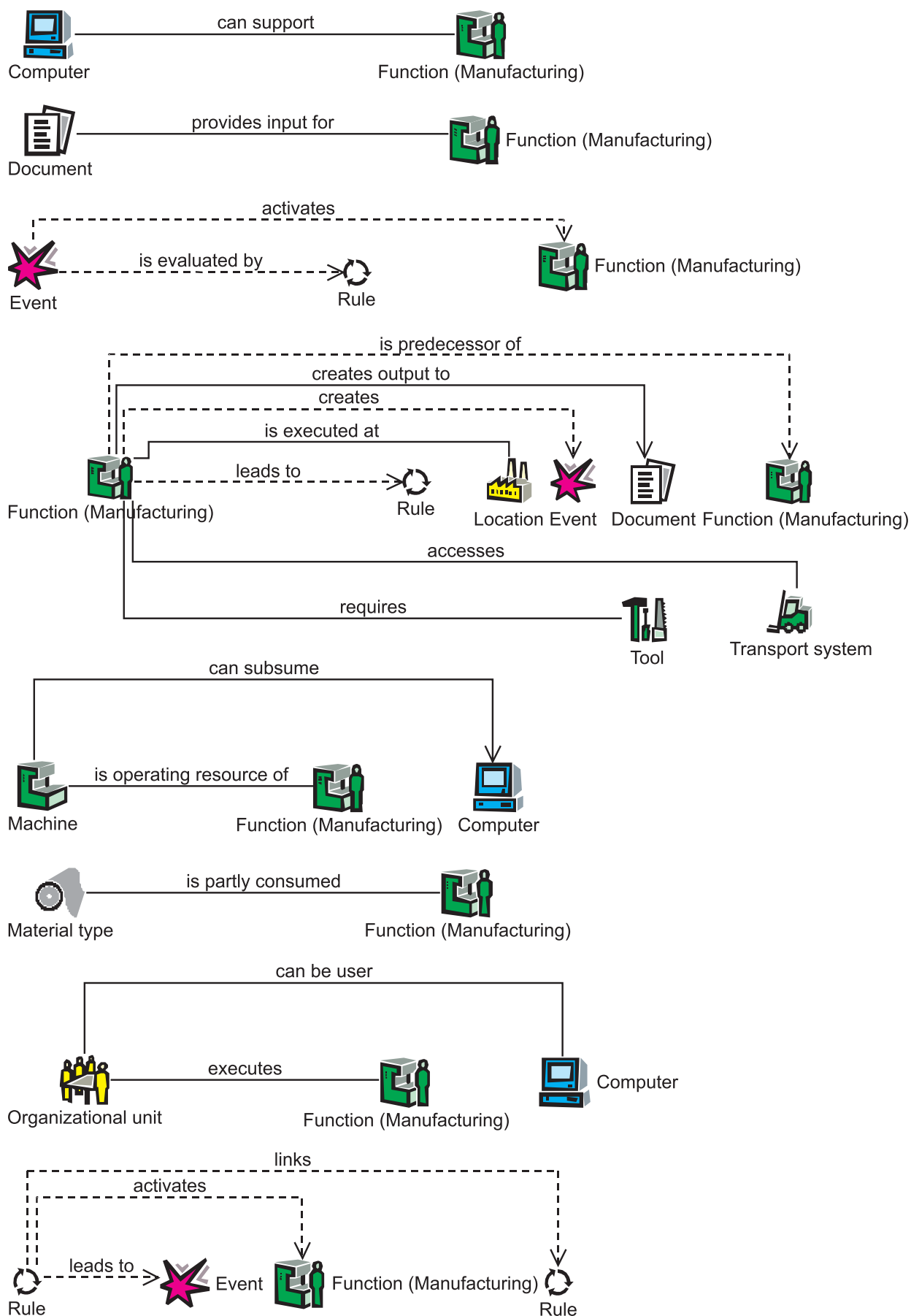


Рис.7.2.22 Объекты и их связи в моделях типа Industrial process

Таблица 7.3.1

| <i>Тип сценария</i>                         | <i>Файловое расширение</i> |
|---|----------------------------|
| Ориентированный на объект (object-oriented) | .rso                       |
| Ориентированный на модель (model-oriented)  | .rsm                       |
| Ориентированный на группу (group-oriented)  | .rsg                       |
| Ориентированный на БД (database-oriented)   | .rsd                       |

ДО Manage Macros. Чтобы открыть это ДО, можно также нажать на кнопку Manage Macros на панели инструментов макроса (Macros toolbar). Заданная по умолчанию панель инструментов макроса содержит две кнопки. На этой панели могут размещаться пиктограммы (символы) макросов, которые были созданы пользователями в текущей установке ARIS. Эти символы выбираются во время создания или редактирования макроса в ДО Select.

Для отображения имен макросов на панели инструментов необходимо выбрать флажок Quick Info в ДО Customize. Для вызова этого ДО достаточно нажать правую кнопку мыши на панели инструментов и выбрать пункт Customize. При нажатии правой кнопки мыши всплывает меню Toolbars, содержащее в качестве пунктов имена доступных пользователю панелей инструментов. Для того, чтобы отобразить (спрятать) панель инструментов, достаточно активизировать (деактивизировать) соответствующий флажок. ДО Customize содержит три вкладки: панели инструментов (Toolbars), команды (Commands), сокращенные клавиши (Shortcut Keys). Работа в этом ДО аналогична работе в соответствующих ДО офисных приложений Word, Excel и др.

ДО Manage Macros позволяет пользователю редактировать и создавать новые макросы. В левой части ДО отображены имена всех макросов, которые доступны в текущей установке. Для редактирования макроса достаточно выбрать его из списка и нажать на кнопку редактирования. Откроется ДО Select Script. Установочные параметры макросов могут быть изменены. Изменения сохраняются при нажатии кнопки ОК. Для удаления выбранного макроса достаточно нажать на кнопку Delete.

Для создания нового макроса необходимо нажать на кнопку New. Откроется ДО мастера макроса Select Script. В этом ДО можно выбрать желаемый сценарий, который будет использован для создания отчета. В раскрывающемся списке Context можно выбрать БД (Database), группу (Group), метод фильтра (Method Filter), модель (Model), объект (Object). В текстовой строке сценария оценки (Script) указывается путь к файлу, в

котором сохранен сценарий и который будет выполняться как макрос. Можно добавить полное имя файла сценария к этому пути или нажать на кнопку поиска и найти сценарий в стандартном ДО Windows Open File. Расширение файла выбранного сценария должно соответствовать выбранной сущности в списке Context. Если, например, выбирается Model, то выбранный сценарий должен иметь файловое расширение rsm. Если сущность и расширение не соответствуют друг другу, то кнопка Next будет недоступна.

В текстовой строке Title появляется описание сценария в виде текста помощи. Если поместить указатель мыши в этой строке, не двигая им в течение нескольких секунд, то будет отображено более подробное описание выбранного сценария в форме быстрой информации (Quick Info). Если текстовая строка пуста или если информация не отображена, то это означает что описание не предусмотрено для выбранного сценария.

В текстовом блоке Macro Name (имя макроса) необходимо ввести имя нового макроса. Можно назначить кнопку для нового макроса. Для этого необходимо открыть раскрывающийся список справа от текстового блока и выбрать желаемый символ.

Можно показать на экране процесс формирования отчета при помощи макроса. Для этого необходимо установить флажок Display Procedure Dialog. После выбора необходимых установок для перехода к следующему ДО необходимо щелкнуть на кнопке Next. Появится ДО Select Script Options, в котором необходимо указать, какой язык БД будет использоваться во время выполнения макроса. Соответствующий раскрывающийся список перечисляет все языки, установленные в текущей операционной системе на компьютере. Для сохранения отчета в файле нужно выбрать флажок создания выходного файла (Create Output File). Для отображения выходного файла после выполнения макроса необходимо установить флажок Display Output File.

В следующей (третьей) строке выбирается выходной формат файла. В раскрывающемся списке предлагаются форматы txt, doc, rtf, xls, html и другие. Не все выходные форматы пригодны для выбранного сценария отчета. Отчет может быть отредактирован в формате doc в Word, и в формате xls в Excel.

В последней текстовой строке вводится путь к папке, в которой сохраняются результаты отчета (файлы выхода). Для выбора желаемой папки выхода можно воспользоваться кнопкой поиска. При этом откроется стандартное ДО Windows Save As, в котором следует определить папку и желаемый файл. Файловое расширение должно соответствовать выбранному выходному фор-

мату. Если, например, выбран формат html, выходной файл должен иметь файловое расширение html. Если файловое расширение и выходной формат не соответствуют друг другу, кнопка Finish будет недоступна.

Для последовательной нумерации каждого создаваемого нового файла выхода при выполнении макроса нужно выбрать флажок Numbering. Тогда, если МоеИмя.rtf – набранное имя для файла отчета, то файл оценки получит имя МоеИмя1.rtf после того, как макрос оценки выполнится впервые. Во время второго выполнения макроса новый выходной файл создастся с именем МоеИмя2.rtf и т. д.

Закончив установки в ДО Select Script Options, нужно щелкнуть на кнопку “Готово”. При этом появится ДО Manage Macros. Для формирования отчета при помощи созданного макроса необходимо выбрать соответствующие сущности, для которых будет создаваться отчет и запустить этот макрос на выполнение.

Рассмотрим теперь процесс создания отчета с помощью мастера отчетов.

Для вызова мастера отчетов необходимо выбрать пункт меню Evaluate/Report, после чего появится ДО мастера отчетов. Это ДО предлагает пользователю выбрать сценарий отчета из списка предлагаемых сценариев. Сценарии выбираются в зависимости от сущности, для которой создается отчет. Раскрывающийся список отображает все сценарии, которые являются допустимыми для выбранной сущности. Для выбранного сценария отображается полный путь к файлу, в котором он находится.

Для создания отчета можно выбрать сценарий из группы других сценариев. Для этого достаточно либо ввести полный путь в текстовое поле, либо найти желаемый файл, используя кнопку поиска. Для получения полного описания выбранного сценария достаточно нажать клавиши Ctrl + F1. После выбора сценария можно перейти к следующему шагу создания отчета, т. е. нажать кнопку “Далее”. Тогда появится ДО выбора параметров файла выхода. Это ДО предлагает пользователю те же возможности выбора, что и ДО мастера макроса.

После запуска созданного макроса или после выбора всех параметров мастера отчетов и запуска его на выполнение появляется ДО, название которого в первом случае соответствует имени макроса, а во втором называется Report. Появление этого ДО свидетельствует о том, что создание отчета началось. Если необходимо прервать процедуру создания отчета, то нужно нажать кнопку Cancel или, если используется макрос, нажать кнопку End на панели инструментов макроса.

На рис.7.3.1 изображен процесс создания отчета, а на рис.7.3.2 представлена диаграмма размещения объектов вокруг этого процесса.

## 7.4. Скрипты анализа моделей и создание новых скриптов

Перечислим и кратко опишем существующие в ARIS скрипты анализа моделей. Эти скрипты используются, как правило, для анализа более сложных типов моделей, чем те, которые допустимы в ARIS 5.0 с простейшим фильтром.

Скрипты 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 анализ моделей, вторые – на анализ объектов. Известны следующие скрипты анализа моделей.

*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 ИС внутри БП. Он может принимать значения от 0 до 1. Чем ниже степень интеграции ИС, тем выше этот коэффициент.

*BSC\_GraphicExport.asm.* Используется для BSC-анализа причинно-следственной диаграммы, дерева функций, организационной схемы и т. д.

*BSC\_PlanActualComparsion.asm.* Применяется для BSC-анализа причинно-следственной диаграммы, дерева функций, модели структуры, организационной схемы и VAD. Выводит результаты сравнения фактических и планируемых показателей достижения целей компании.

*FunctionClass.asm.* Применяется для анализа всех моделей, содержащих функции. Проводится группировка всех функций выбранных моделей согласно значению атрибута определенного типа.

*MediaBreak.asm.* Используется для разновидностей моделей eEPC, производственного и офисного БП, UML-диаграммы действий, PCD и PCD с потоком материалов. В ходе анализа определяется коэффициент, отражающий степень информационных разрывов в БП. Этот коэффициент может принимать значения от 0 до 1. Чем ниже степень информационной интеграции в БП, тем выше значение коэффициента.

*ModelComparsion.asm.* Применяется для сравнения моделей двумя способами. В первом (сравнение существования) выявляется наличие или отсутствие одинаковых объектов в сравниваемых моделях. В качестве критериев идентичности могут выступать некоторые атрибуты (имя, идентификатор, описание объекта и т. д.). Во втором сравниваются атрибуты объектов с идентичными именами, находящимися в различных моделях.

*OrgChange.asm.* Применяется для разновидностей модели eEPC, производственного и офисного БП, UML-



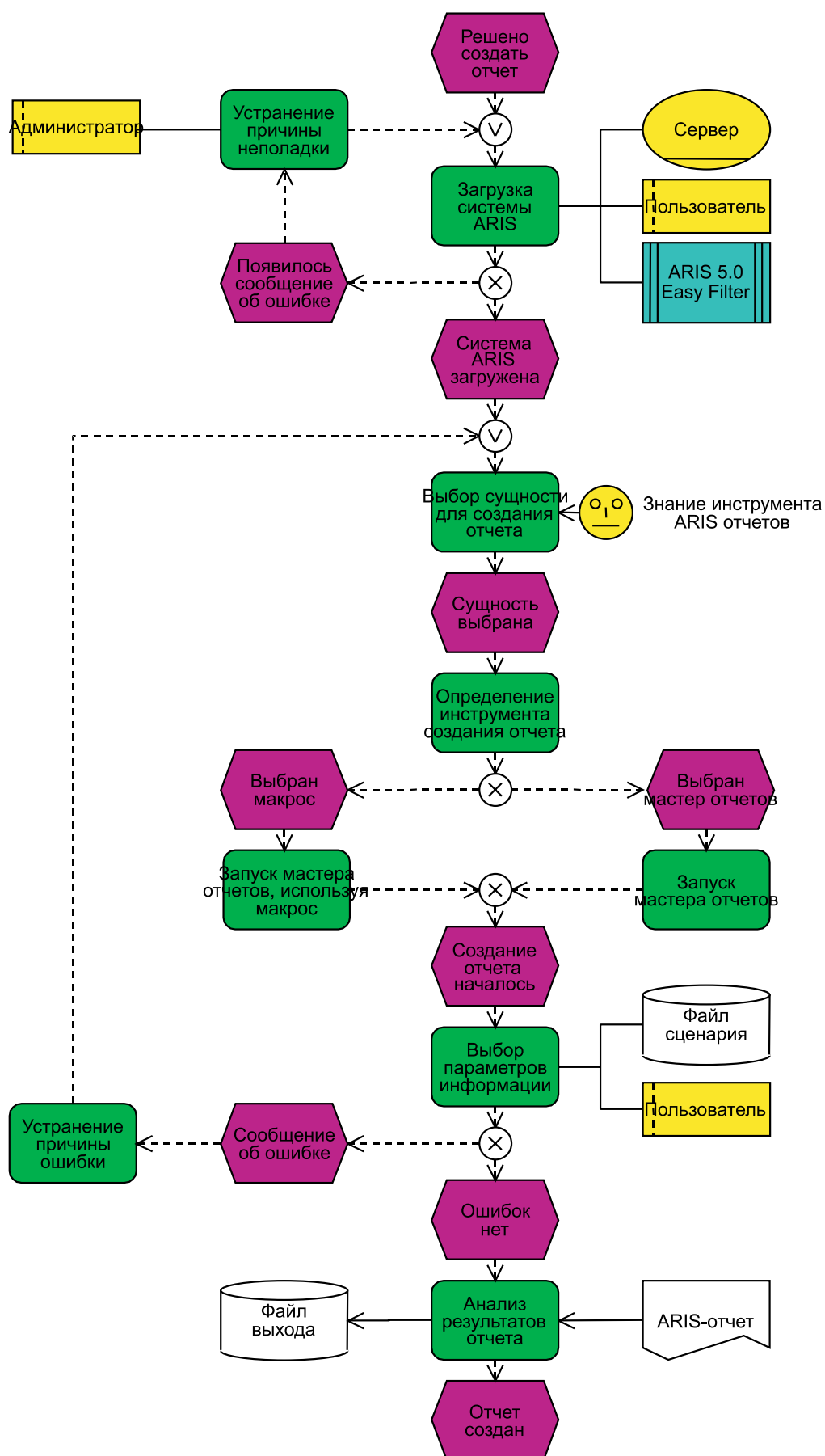


Рис.7.3.1 Процесс создания отчета

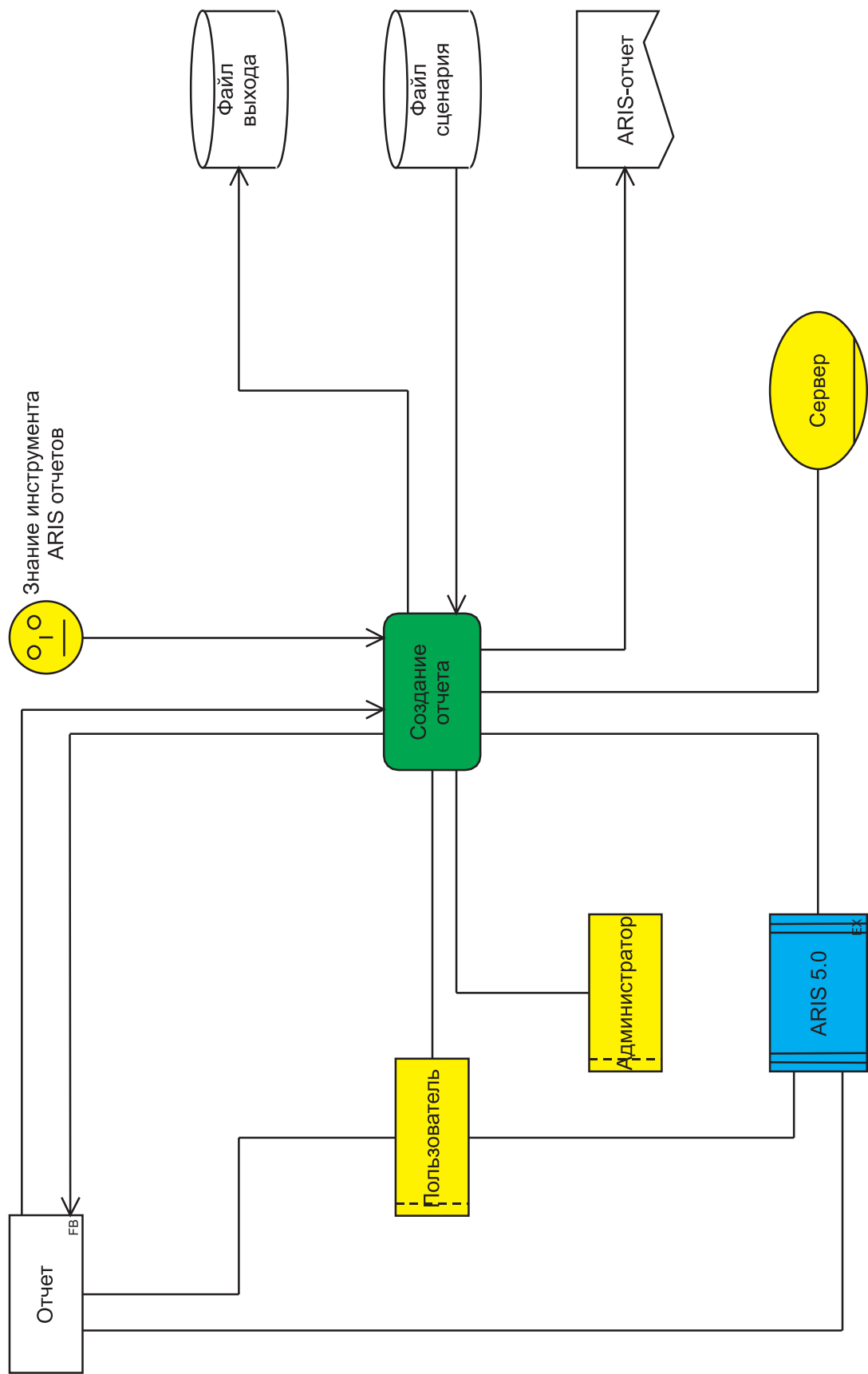


Рис.7.3.2 Диаграмма размещения объектов вокруг функции "Создание отчета"

диаграммы действий, PCD и PCD с потоком материалов. Определяется степень ориентированности на БП. Расчет основывается на количестве переходов ответственности от одного исполнителя к другому.

*VariantComparision.asm*. Применяется для сравнения только двух моделей – мастер-модели и ее варианта.

Оригинальные или подобные этим скрипты анализа, а также скрипты подготовки отчетов по моделям и результатам их анализа может создать любой пользователь самостоятельно. Для этого существуют соответствующие механизмы.

В процессе создания скрипта рекомендуется в первую очередь описать цели его создания, определить будущих пользователей, проанализировать требования пользователей к отчету, оценить возможности ARIS по выводу информации в нужном виде, оценить и скорректировать требования к моделям, с которыми создаваемый скрипт будет работать и т. д. Все здесь перечисленное относится к этапу анализа требований к будущему скрипту. Результаты этих требований должны отразиться в техническом задании на разработку скрипта. В нем описывается выводимая информация, перечисляются ARIS-сущности (модели, объекты, связи и т. д.), которые будут анализироваться, определяются формы представления данных (структурированный текст, таблицы, графика модели и т. д.). В задании на программирование необходимо описать структуру документов 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.

Создать скрипт можно с помощью мастера скриптов (*Script Wizard*). Он позволяет создать структуру (шаблон) скрипта, где указывается, какие модели, объекты, связи и атрибуты будут обрабатываться этим скриптом и какая фильтрация и сортировка будет использоваться. Этот мастер позволяет реализовать желаемую структуру вывода информации и в целом сформировать структуру новой (будущей) программы. Изменить и русифицировать скрипт можно с помощью *ARIS Script Editor*.

Структура и состав выводимой в отчет информации могут быть полностью определены пользователем. Для этого необходимо разработать собственный скрипт при помощи редактора скриптов (модуль *ARIS Script Editor*).

*Script Wizard* запускается через пункт меню *File/New*. Выбрав в появившемся окне пункт меню *Evaluation Script*, откроется ДО (*Script Wizard – Select Basic Settings*) задания основных параметров будущего скрипта. В первом списке (*For which component do you wish to create the script?*) этого ДО необходимо указать тип скрипта *Report*

(*Отчет*). Во втором списке (*For which context do you wish to create the script?*) указываются ARIS-сущности, для работы с которыми создается скрипт (для объектов, моделей, папок и т. д.). В текстовом поле *Script file name* задается имя файла, в котором будет сохранен создаваемый скрипт. Для перехода к дальнейшим шагам создания скрипта необходимо в области *Generation (Генерация)* выбрать опцию *Generate Code (Генерировать код)*. Если выбрать опцию *Create Empty Script (Создать пустой скрипт)*, то в программе скрипта будут сгенерированы только заголовки. Нажав в этом ДО кнопку *Properties (Свойства)*, можно задать заголовок (название скрипта), автора (создателя, разработчика) скрипта, название компании разработчика, краткое описание и другую информацию для создаваемого скрипта. В следующем ДО (*Script Wizard – Select Filter Settings*) указывается фильтр, через который скрипт будет обращаться к БД, а также типы объектов и моделей, для которых рекомендуется применять новый скрипт. В следующем ДО (*Script Wizard – Define Basic Layout*) можно определить колонтитулы будущего отчета.

В следующем ДО (*Script Wizard – Define Procedural Structure*) формируется иерархическая структура выводимой в отчет информации. Напомним, что ARIS использует объектную БД, где объекты являются экземплярами классов. Описание этих классов можно найти в контекстной справке по системе ARIS (раздел *ARIS Script Editor*). В правой части ДО (*Script Wizard – Define Procedural Structure*) представлен список классов и их свойств. После добавления в программу из этого списка класса или его свойства автоматически генерируются соответствующие операторы кода. Если добавляемый класс или свойство является списком (они помечены символом из двух ромбиков), то в отчет выводится содержимое списка. Для упорядочивания этого списка нужно в этом ДО нажать *Options (Опции)* и в появившемся ДО сортировки списка указать от одного до трех критериев его сортировки.

Сформировав таким образом структуру скрипта и нажав кнопку *Готово*, можно записать текст программы в файл. При отсутствии ошибок выдается сообщение об успешном окончании создания программы.

*Script Wizard* позволяет изменять созданный скрипт. Для этого необходимо через меню *File/Open* вызвать ДО, в котором выбрать пункт *Evaluation Script*. Далее в ДО *Select Script* выбрать файл изменяемого скрипта. В ДО *Script Wizard – Select Processing Type* следует задать способ редактирования скрипта. Редактирование при помощи *Script Wizard* задается опцией *In Script Wizard*, а при помощи *ARIS Script Editor* – опцией *In Script Editor*. Редактирование

с помощью *ARIS Script Editor* выбирается в том числе, если необходимые изменения выходят за рамки возможностей *Script Wizard*. В ДО *Script Wizard – Select Processing Type* нажав кнопку *Properties*, можно изменить заголовок, автора, название компании, описание программы и т. д.

Следует иметь в виду, что если разрабатываемый скрипт был открыт и сохранен при помощи *ARIS Script Editor*, то этот скрипт уже нельзя будет открыть при помощи *Script Wizard*.

*ARIS Script Editor* предназначен для разработки, создания и редактирования авторских скриптов анализа и подготовки отчетов. В заголовке экрана этого редактора (рис. 7.4.1) видно полное имя файла, в котором хранится программа. Затем следует главное меню и панель инструментов. В верхней части окна с текстом кода находится два раскрывающихся списка, где правый отражает структуру программы (список всех процедур и функций, блок глобальных объявлений переменных) и позволяет перемещаться между структурными блоками программы. В нижней части окна имеются закладки для просмотра текста программы (*Script*), макета отчета (*Layout* только для чтения), верхнего (*Header*) и нижнего (*Footer*) колонтитулов.

В тексте программы различными цветами выделены семантически различные участки кода. Комментарий – зеленый. Команды языка ARIS-Basic – темно-синий. Команды Visual Basic – синий. Структуры языка (определенные пользователем объекты их методы, константы и т. д.) – черный. Строки с ошибками – красный. Контрольные термины – бирюзовый.

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. Ниже перечислены те ARIS-объекты и их свойства, которые используются в пяти скриптах, написанных в качестве авторских (студенческих) программ для выполнения элементарных операций с различными типами моделей, построенных в ARIS 5.0 с простейшим фильтром.

*OutPut.Init(long outputType, VARIANT localeID)* – инициализирует и устанавливает формат файла вывода. При необходимости определяет язык вывода. Здесь *outputType* – формат файла вывода. Можно использовать форматы: *OUTRTF*, *OUTTEXT*, *OUTWORD*, *OUTEXCEL*, *OUTHTML*, *OUTTABLE*. *localeID* – язык файла вывода. Если язык не задан, то используется язык интерфейса ARIS.

*OutPut.DefineF(sStyleSheetName, sFontName, long nFontSize, long fontColor, long bkColor, long nFormat, long mm\_indLeft, long mm\_indRight, long mm\_distTop, long*

*mm\_distBottom, long mm\_indFirstLine, double dLineSpacing)* – определяет шаблон стиля (для документов MS Office), где *sStyleSheetName* – название шаблона, *sFontName* – название шрифта, *nFontSize* – размер шрифта, *fontColor*, *bkColor* – цвет шрифта и фона, *nFormat* – форматирование текста, *mm\_indLeft*, *mm\_indRight*, *mm\_distTop*, *mm\_distBottom* – отступы слева, справа, сверху и снизу, *mm\_indFirstLine* – абзацный отступ, *dLineSpacing* – высота строки.

*OutPut.Output(VARIANT sVarText, LPCTSTR fontName, long fontSize, long fontColor, long bkColor, long nFormat, long mmIndent)* – выводит текст с заданным форматированием в объект вывода, где *sVarText* – текст для вывода. Остальные параметры аналогичны *OutPut.DefineF*.

*OutPut.OutputLnF(VARIANT sVarText, LPCTSTR sStyleSheetName)* – выводит текст с форматированием по заданному шаблону в объект вывода, где *sVarText* – смотрите выше, *sStyleSheetName* – название шаблона.

*OutPut.WriteReport(LPCTSTR strPath, LPCTSTR strFileName)* – записывает отчет, созданный командами вывода в формате, заданном *OutPut.Init* в файл с именем *strFileName*, с расположением *strPath* (путь к файлу).

*BaseList.Get(long index)* – возвращает элемент списка в позиции *index* (начиная с нуля).

*BaseList.Add(LPDISPATCH pDisp, VARIANT nPos)* – добавляет элемент в список, где *pDisp* – добавляемый элемент, *nPos* – позиция добавляемого элемента (необязательный параметр), если она не указана, то элемент добавляется в конец списка.

*BaseList.Sort(short sortKrit0, short sortKrit1, short sortKrit2, long localeID)* – сортирует список по нескольким критериям. Если критериев меньше трех, то для оставшихся должно стоять *SORT\_NONE*.

*long BaseList.Count()* – возвращает количество элементов в списке.

*ObjOcc.ObjDef()* – возвращает определение объекта как *ObjDef*.

*ObjOcc.Attribute(long attrTypeNum, long localeID)* – возвращает атрибут *attrTypeNum* на языке *localeID* как *Attr*. Если язык не указан, то используется язык системы ARIS.

*long ObjOcc.InDegree(short nStructure)* – возвращает количество приходящих связей объекта, где *nStructure* – *EDGES\_ALL* (все связи), *EDGES\_STRUCTURE* (структурные связи), *EDGES\_NONSTRUCTURE* (неструктурные связи).

*ObjOcc.Cxns(VARIANT nInOut, VARIANT nStructure)* – возвращает список связей объекта как *CxnsOccList*, где *nInOut* – *EDGES\_ALL* (все связи), *EDGES\_IN* (приходящие связи), *EDGES\_OUT* (исходящие связи), *nStructure* – смотрите выше.

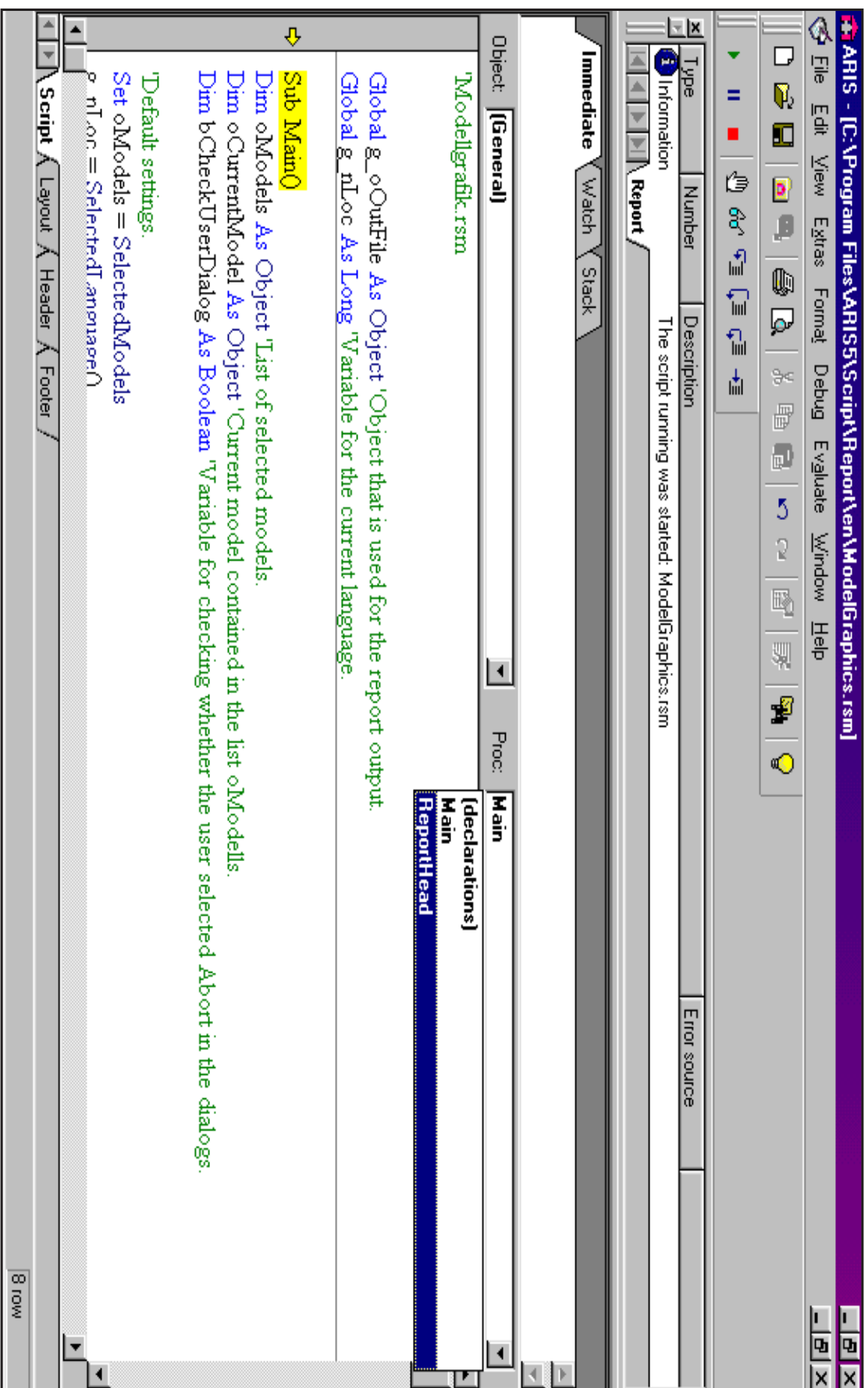


Рис. 7.4.1 Экран редактора скриптов ARIS

*bool Attr.IsMaintained()* – определяет, поддерживает-ся данный атрибут и содержит ли он какие-либо данные (непустую строку).

*Attr.GetValue(BOOL bRemoveLineBreaks)* – возвращает значение атрибута. Если *bRemoveLineBreaks* – истина, то символы перевода строки удаляются из возвращаемого значения.

*Attr.MeasureValue(BOOL bRemoveLineBreaks)* – возвращает значение атрибута с единицами измерения. Если *bRemoveLineBreaks* – истина, то удаляет символы перевода строки.

*Attr.SetValue(LPCTSTR sValue, long nMeasureUnitTypeNum)* – присваивает новое значение атрибуту, где *sValue* – присваиваемое значение, *nMeasureUnitTypeNum* – номер единиц измерения.

*CxnOcc.TargetObjOcc()* – возвращает объект, на который указывает связь, как *ObjOcc*.

## 7.5. Программы обработки моделей технических терминов

Текст программы первого учебного скрипта для работы с произвольной моделью технических терминов, построенной в ARIS 5.0 с простейшим фильтром, изображен на рис.7.5.1. Эта программа предназначена для создания отчета, содержащего список определений технических

терминов выбранной модели. Для корректной работы программы необходимо выбрать модель, содержащую хотя бы один технический термин, и в ДО запуска скриптов выбрать документ Word в качестве формата файла отчета. Технические термины, для которых предполагается получить список определений, должны иметь имя (непустую строку) и заполненный атрибут *Description\Definition* (*Описание\Определение*). Текст определения технического термина берется из этого атрибута. В процессе работы программы формируется отчет, структура которого представлена в таблице 7.5.1.

Таблица 7.5.1

| Строка      | Содержание строки   |
|-------------|---|
| 1           | Определения   |
| 2           | понятий из модели   |
| 3           | Тип модели:<br><тип выбранной модели>                           |
| 4           | Название модели:<br><Название выбранной модели>                 |
| 5           | <пустая строка>   |
| 6           | <пустая строка>   |
| 7, 8 и т.д. | <Название технического термина>.<br><Определение этого термина> |

Если выбрать модель технических терминов на рис.7.5.2 и запустить эту программу, то в результате сформируется следующий отчет.

### Определения понятий из модели Тип модели : Technical terms model Название модели : Определения AC

**Автоматизированная система (AC).** Система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.

**Автоматизированное рабочее место (АРМ).** Программно-технический комплекс AC, предназначенный для автоматизации деятельности определенного вида.

**Адаптивность AC.** Способность AC изменяться для сохранения своих эксплуатационных показателей в заданных пределах при изменениях внешней среды.

**Алгоритм функционирования AC.** Алгоритм, задающий условия и последовательность действий компонентов AC при выполнении ею своих функций.

**Внешняя информационная база.** Часть информационной базы AC, представляющая собой совокупность документов, предназначенных для непосредственного восприятия человеком без применения средств вычислительной техники.

**Входная информация AC.** Информация, поступающая в AC в виде документов, сообщений, данных, сигналов, необходимая для выполнения функций AC.



```

Global OutFile As Object ' Объект для вывода результатов в файл

Sub Main
    Dim Models As Object ' Models хранит выбранную модель
    Dim Terms As Object ' Terms хранит список определений терминов
    Set Models = SelectedModels ' Определение выбранной модели
    Set Terms = CreateObject("ARIS.ObjOccList.5.0") ' Создание списка рассматриваемых объектов
    Set OutFile = CreateObject("ARIS.Output.5.0") ' Создание объекта для вывода списка определений
    OutFile.Init(OUTWORD, SelectedLanguage) ' Инициализация выводного документа Word (OUTWORD) с
                                           выбранным языком (SelectedLanguage)

    GetTerms Models, Terms ' Выделение всех объектов из модели
    Terms.Sort(AT_NAME, SORT_NONE, SORT_NONE, SelectedLanguage)
    ' Сортировка по именам объектов (AT_NAME) на основе выбранного языка
    OutTerms Terms, Models ' Формирование отчета и вывод его в файл
End Sub

Sub GetTerms (Models As Object, ByRef Terms As Object)
    Dim TermsObjOccs As Object ' Рабочий объект
    Dim i As Long
    Set TermsObjOccs = Models.Get(0).ObjOccList ' Выбираем все элементы модели
    For i = 0 To TermsObjOccs.Count()-1 ' Перебор всех объектов
        If TermsObjOccs.Get(i).ObjDef().Attribute(AT_DESC, SelectedLanguage).IsMaintained() Then
            ' Выбираются только те объекты, для которых существует описание (AT_DESC)
            Terms.Add(TermsObjOccs.Get(i)) ' Передает объект в переменную Terms
        End If
    Next i
End Sub

Sub OutTerms (Terms As Object, Models As Object) ' Выводит список определений в файл
    Dim i As Long
    Dim fname As String ' Строка для хранения имени объекта
    ' Определение стилей: REPORT1 - для заголовка, REPORT2 - для справочной информации
    ' REPORT4 - для описаний терминов
    OutFile.DefineF("REPORT1", "Arial", 16, C_BLACK, COLOR_TRANSPARENT, FMT_CENTER, 0, 0, 0, 0, 1)
    OutFile.DefineF("REPORT2", "Arial", 14, C_BLACK, COLOR_TRANSPARENT, FMT_CENTER, 0, 0, 0, 0, 1)
    OutFile.DefineF("REPORT4", "Times New
                    Roman", 12, C_BLACK, COLOR_TRANSPARENT, FMT_JUSTIFY, 0, 0, 0, 0, 1)
    ' Определение полей (сверху, слева, справа, снизу)
    OutFile.TopMargin = 35 : OutFile.LeftMargin = 35 : OutFile.RightMargin = 25 : OutFile.BottomMargin = 30
    ' Вывод названия отчета и модели
    OutFile.OutputLnF("Определения", "REPORT1") : OutFile.OutputLnF("понятий из модели", "REPORT2")
    OutFile.OutputLnF("", "REPORT4") : OutFile.OutputLnF("Тип модели : " +
                                                         Models.Get(0).Type(), "REPORT2")
    OutFile.OutputLnF("Название модели : " + Models.Get(0).Name(SelectedLanguage), "REPORT2")
    OutFile.OutputLnF("", "REPORT4") : OutFile.OutputLnF("", "REPORT4")
    For i = 0 To Terms.Count()-1 ' Перебор всех объектов
        fname = Terms.Get(i).ObjDef().Attribute(AT_NAME, SelectedLanguage).GetValue(True)
        ' Вывод названия термина (курсивом) и его определения
        OutFile.Output(fname + " ", "Times New
                        Roman", 12, C_BLACK, COLOR_TRANSPARENT, FMT_ITALIC, 0)
        OutFile.OutputLnF(Terms.Get(i).ObjDef().Attribute(AT_DESC, SelectedLanguage).GetValue(True), "REPO
                                                                RT4")
        ' Пустая строка между определениями
        OutFile.OutputLnF("", "REPORT4")
    Next i
    OutFile.WriteReport(SelectedPath, SelectedFile) ' Запись сформированного отчета в файл
End Sub

```

Рис.7.5.1 Программа вывода определений объектов в модели технических терминов

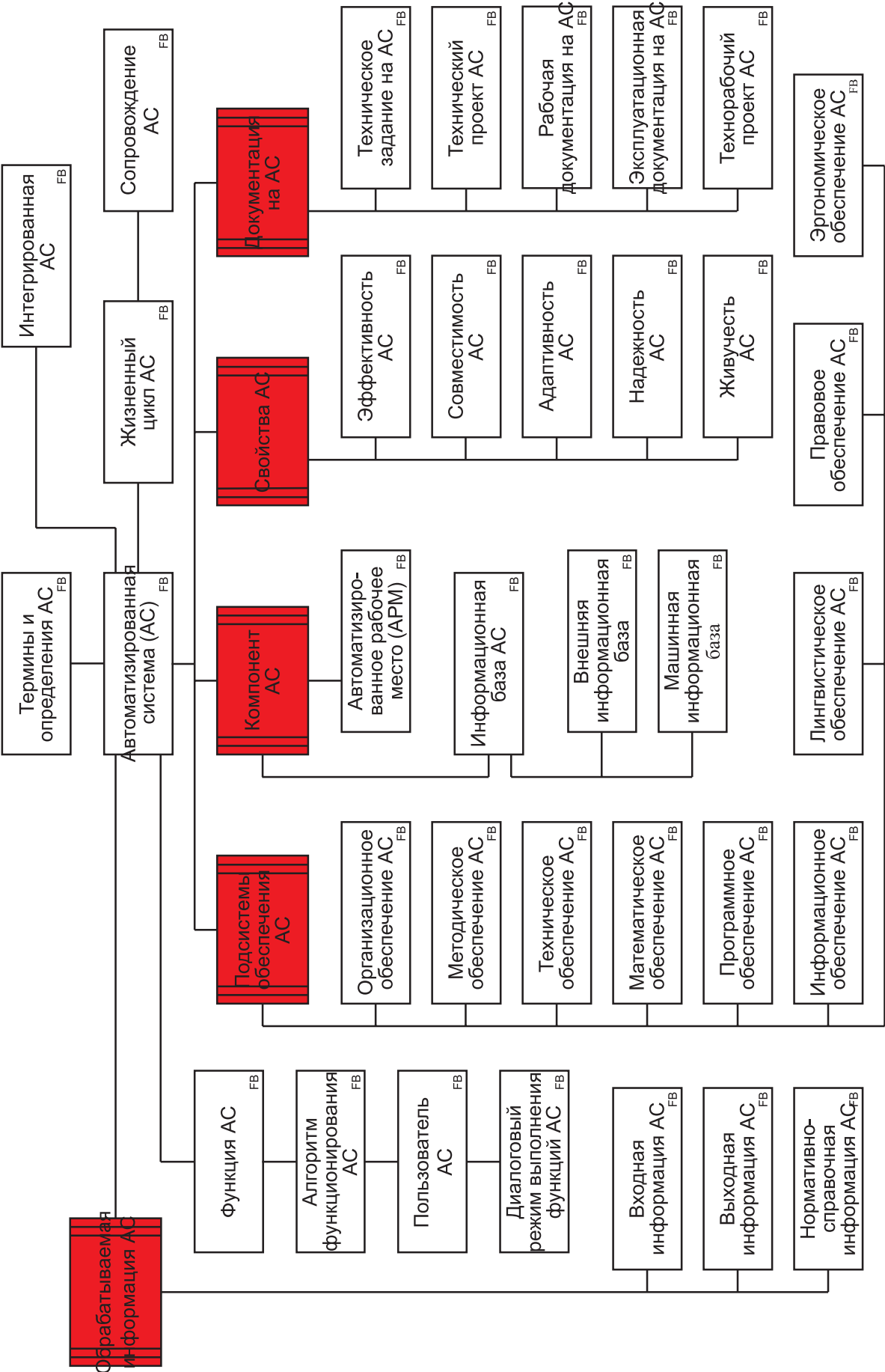


Рис. 7.5.2 Диаграмма модели технических терминов для терминов и понятий АС

*Выходная информация АС.* Информация, получаемая в результате выполнения функций АС и выдаваемая на объект ее деятельности пользователю или в другие системы.

*Диалоговый режим выполнения функций АС.* Режим выполнения функции АС, при котором пользователь АС управляет решением задачи, изменяя ее условия и/или порядок функционирования АС на основе оценки информации, представляемой ему техническими средствами АС.

*Живучесть АС.* Свойство АС, характеризующееся способностью выполнять установленный объем функций в условиях воздействий внешней среды и отказов компонентов системы в заданных пределах.

*Жизненный цикл АС.* Совокупность взаимосвязанных процессов создания и последовательного изменения состояния АС от формирования исходных требований к ней до окончания эксплуатации и утилизации комплекса средств автоматизации АС.

*Интегрированная АС.* Совокупность двух или более взаимоувязанных АС, в которой функционирование одной из них зависит от результатов функционирования другой (других) так, что эту совокупность можно рассматривать как единую АС.

*Информационная база АС.* Совокупность упорядоченной информации, используемой при функционировании АС.

*Информационное обеспечение АС.* Совокупность форм документов, классификаторов, нормативной базы и реализованных решений по объемам, размещению и формам существования информации, применяемой в АС при ее функционировании.

*Компонент АС.* Часть АС, выделенная по определенному признаку или совокупности признаков и рассматриваемая как единое целое.

*Лингвистическое обеспечение АС.* Совокупность средств и правил для формализации естественного языка, используемых при общении пользователей и эксплуатационного персонала АС с комплексом средств автоматизации при функционировании АС.

*Математическое обеспечение АС.* Совокупность математических методов, моделей и алгоритмов, примененных в АС.

*Машинная информационная база.* Часть информационной базы АС, представляющая собой совокупность используемой в АС информации на носителях данных.

*Методическое обеспечение АС.* Совокупность документов, описывающих технологию функционирования АС, методы выбора и применения пользователями технологических приемов для получения конкретных результатов при функционировании АС.

*Надежность АС.* Комплексное свойство АС сохранять во времени в установленных пределах значения всех параметров, характеризующих способность АС выполнять свои функции в заданных режимах и условиях эксплуатации.

*Нормативно-справочная информация АС.* Информация, заимствованная из нормативных документов и справочников и используемая при функционировании АС.

*Организационное обеспечение АС.* Совокупность документов, устанавливающих организационную структуру, права и обязанности пользователей и эксплуатационного персонала АС в условиях функционирования, проверки и

обеспечения работоспособности АС.

*Пользователь АС.* Лицо, участвующее в функционировании АС или использующее результаты ее функционирования.

*Правовое обеспечение АС.* Совокупность правовых норм, регламентирующих правовые отношения при функционировании АС и юридический статус результатов ее функционирования.

*Программное обеспечение АС.* Совокупность программ на носителях данных и программных документов, предназначенная для отладки, функционирования и проверки работоспособности АС.

*Рабочая документация на АС.* Комплект проектных документов на АС, разрабатываемый на стадии “Рабочая документация”, содержащий взаимоувязанные решения по системе в целом, ее функциям, всем видам обеспечения АС, достаточные для комплектации, монтажа, наладки и функционирования АС, ее проверки и обеспечения работоспособности.

*Совместимость АС.* Комплексное свойство двух или более АС, характеризующее их способностью взаимодействовать при функционировании (включающее техническую, программную, информационную, организационную и лингвистическую совместимость).

*Сопровождение АС.* Деятельность по оказанию услуг, необходимых для обеспечения устойчивого функционирования или развития АС.

*Технический проект АС.* Комплект проектных документов на АС, разрабатываемый на стадии “Технический проект”, утвержденный в установленном порядке, содержащий основные проектные решения по системе в целом, ее функциям и всем видам обеспечения АС и достаточный для разработки рабочей документации на АС.

*Техническое задание на АС.* Документ, оформленный в установленном порядке и определяющий цели создания АС, требования к АС и основные исходные данные, необходимые для ее разработки, а также план-график создания АС.

*Техническое обеспечение АС.* Совокупность всех технических средств, используемых при функционировании АС.

*Технорабочий проект АС.* Комплект проектных документов АС, утвержденный в установленном порядке и содержащий решения в объеме технического проекта и рабочей документации на АС.

*Функция АС.* Совокупность действий АС, направленная на достижение определенной цели.

*Эксплуатационная документация на АС.* Часть рабочей документации на АС, предназначенная для использования при эксплуатации системы, определяющая правила действия персонала и пользователей системы при ее функционировании, проверке и обеспечении ее работоспособности.

*Эргономическое обеспечение АС.* Совокупность реализованных решений в АС по согласованию психологических, психофизиологических, антропометрических, физиологических характеристик и возможностей пользователей АС с техническими характеристиками комплекса средств автоматизации АС и параметрами рабочей среды на рабочих местах персонала АС.

*Эффективность АС.* Свойство АС, характеризующее степень достижения целей, поставленных при ее создании.

Текст программы второго скрипта для работы со специализированной моделью технических терминов изображен на рис.7.5.3 (занимает две страницы). Эта программа вычисляет величину переменных затрат на производство одного продукта. Для этого суммируются все указанные в модели переменные издержки. Для работы программы необходимо выбрать модель технических терминов, построенную в соответствии с определенными соглашениями, и указать документ Word как формат файла вывода.

Структура модели, которую анализирует рассматриваемая программа, поясняется на рис.7.5.4. В этой модели для каждой затратной статьи, описанной в модели кластером, должны быть определены конкретные числовые значения всех составляющих этой статьи и занесены в атрибуты объектов (технических терминов) модели. В атрибут *Description\Definition* заносится стоимость одной единицы расходных материалов (комплектующих изделий, энергоносителей и т. д.), а в атрибут *Remark\Example* – количество материалов (комплектующих изделий, энергоносителей и т. д.), необходимых для производства одной единицы готовой продукции и т. д. Также необходимо заполнить атрибут *Description\Definition* для объекта “*Стоимость возвратных средств*” и указать количество видов издержек в каждой категории затрат (атрибут *Remark\Example* у кластеров).

В процессе работы этой программы формируется отчет, содержащий сообщение об ошибке, если структура выбранной модели не соответствует описанным соглашениям (требованиям). Если программа выполняет все необходимые вычисления без сбоев, то появляется сообщение о ее успешном выполнении.

В результате работы программы в атрибут *Description\Definition* объекта “*Переменные затраты на один продукт*” заносится вычисленная величина переменных затрат на один продукт, а в кластеры, обозначающие категории затрат, заносится вычисленная сумма затрат по данной категории (атрибут *Description\Definition*).

Программа на рис.7.5.3 использована при моделировании производственно-сбытовой фирмы, выпускающей мужские сорочки. Соответствующие расчетные формулы изложены в последней главе.

## 7.6. Программа анализа функционального дерева

Для перехода от единых тарифных ставок к расчету зарплаты участников БП в зависимости от их вклада в реализацию БП и величины полученной прибыли можно поступить следующим образом: вычислить вы-

ручку и зафиксировать процент участия исполнителей функций в цепи выполненного БП. Здесь имеется в виду участие в достижении конечной цели – получение прибыли. Если эти проценты заносятся (формируются вычислительным путем) в атрибут *Avg. personnel costs* (Средние трудовые издержки) каждой функции в пройденных цепях БП, то они автоматически запишутся в функции процессно-ориентированного дерева, соответствующего eEPC-диаграмме рассматриваемого БП. Программа, текст которой представлен на рис.7.6.1, анализируя это дерево, вычислит абсолютную зарплату исполнителей каждой функции в цепи БП.

Программа на рис.7.6.1 анализирует процессно-ориентированное дерево функций, т. е. модель типа *Function tree*, и рассчитывает распределение зарплаты по исполнителям каждой функции с учетом их вклада (в процентах) в реализацию БП. Перед использованием этой программы общая зарплата должна быть занесена в атрибут *Avg. other costs* самой первой (комплексной) функции дерева. Название этой функции является названием выполненного (экземпляра) БП с определенным фондом заработной платы исполнителей этого БП. Для всех остальных функций дерева должны быть заполнены атрибуты *Avg. personnel costs* (Средние трудовые издержки) значениями процентного вклада исполнителя данной функции в выполнение процесса (функции) высшего уровня.

В ходе выполнения этой программы вычисляются и обновляются значения атрибутов *Avg. other costs* для всех функций, входящих в детализируемую функцию. В итоге атрибут *Avg. other costs* каждой функции содержит абсолютное значение зарплаты, отнесенной к этой функции.

Программа разносит зарплату исполнителя функции верхнего уровня по зарплатам исполнителей каждой функции нижнего уровня в соответствии с их процентным вкладом в выполнение процесса. Она итерационно (от уровня к уровню) заносит в атрибут *Avg. other costs* каждой функции нижнего уровня произведение содержимого атрибута *Avg. other costs* функции верхнего уровня на сотую часть процента, содержащегося в атрибуте *Avg. personnel costs* функции нижнего уровня.

Алгоритм вычислительной работы программы на рис.7.6.1 представлен в диаграмме типа *Program flow-chart* (PF) (блок-схема программы), изображенной на рис.7.6.2.

## 7.7. Программа обработки организационной диаграммы

Текст программы скрипта для работы с произвольной организационной диаграммой изображен на рис.7.7.1 (занимает три страницы). Эта программа предназначена для



Public OutFile As Object ‘ Объект для вывода результатов в файл

Sub Main

```

    Dim Models As Object ‘ Models хранит выбранную модель
    Dim CIList As Object ‘ CIList хранит список кластеров (категорий затрат)
    Dim Incorrect As Boolean ‘ Признак нарушения структуры модели
    Dim VC As Double ‘ Переменные затраты (результат работы программы)
    Set Models = SelectedModels ‘ Определение выбранной модели
    Set CIList = CreateObject(“ARIS.ObjOccList.5.0”) ‘ Создание списка кластеров
    Set OutFile = CreateObject(“ARIS.Output.5.0”) ‘ Создание и инициализация файла отчета
    OutFile.Init(OUTTEXT,SelectedLanguage)
    OutFile.DefineF(“A”,“Arial”,10,C_BLACK,COLOR_TRANSPARENT,FMT_LEFT,0,0,0,0,0)
    Incorrect = False ‘ Предполагаем, что структура модели правильная
    If Models.Get(0).Type <> “Technical terms model” Then
        OutFile.OutputLnF(“Ошибка : выбранная модель не является моделью технических
терминов”,”A”)
    Else ‘ Проверка типа модели (программа работает только с моделью технических терминов)
        FindFirst Models,CIList,VC,Incorrect ‘ Проверка структуры модели
        ‘ Если во время работы подпрограммы появилась ошибка, то прекратить выполнение
        If Incorrect Then
            OutFile.OutputLnF(“Ошибка : неверная структура модели”,”A”)
        Else
            Calc CIList,VC,Incorrect ‘ Суммирует все издержки
            If Incorrect Then ‘ Если возникла ошибка, то прекратить выполнение
                OutFile.OutputLnF(“Ошибка : количество видов затрат не соответствует
заданному”,”A”)
            Else
                ‘ Если ошибок нет, то записать величину переменных затрат
                CIList.Get(0).ObjDef().Attribute(AT_DESC,SelectedLanguage).SetValue(VC,-1)
                OutFile.OutputLnF(“Анализ успешно завершен”,”A”)
            End If
        End If
    End If
    OutFile.WriteReport(SelectedPath,SelectedFile) ‘ Вывод сообщения об ошибках
End Sub

```

```

Sub FindFirst(Models As Object, ByRef CIList As Object, ByRef VC As Double, ByRef Incorrect As Boolean)
    Dim i As Long
    Dim DummyObjs As Object ‘ Временная переменная для хранения списка объектов
    Dim DummyCxns As Object ‘ Временная переменная для хранения списка связей
    Dim Check As Long ‘ Временная переменная для хранения количества связей
    Dim CCheck As Boolean ‘ Признак правильности модели
    Set DummyObjs = CreateObject(“ARIS.ObjOccList.5.0”) ‘ Создаем список объектов
    Set DummyCxns = CreateObject(“ARIS.CxnOccList.5.0”) ‘ Создаем список связей
    ‘ Выбираем все технические термины
    Set DummyObjs = Models.Get(0).ObjOccListFilter(“”,SelectedLanguage,OT_TECH_TRM)
    If DummyObjs.Count()>0 Then
        CCheck = False
        ‘ Поиск элемента “Переменные затраты на один продукт”
        For i = 0 To DummyObjs.Count()-1
            ‘ Поиск ведется по числу связей (все прочие технические термины имеют только
одну связь)
            Check = DummyObjs.Get(i).OutDegree(EDGES_ALL)
            If Check > 1 Then
                CIList.Add(DummyObjs.Get(i),0) : CCheck = True ‘ Добавляем его в CIList
            Else ‘ Если предыдущий блок выполнен правильно, то можно выполнять следующий
                VC = VC + Check
            End If
        Next i
        If CCheck Then
            ‘ Определяет список связей от нулевого объекта
            Set DummyCxns = CIList.Get(0).Cxns(EDGES_OUT,EDGES_ALL)
            ‘ DummyObjs теперь используется для хранения одного объекта
            Set DummyObjs = Nothing : Set DummyObjs = CreateObject(“ARIS.ObjOcc.5.0”)
        End If
    End If
End Sub

```

Рис.7.5.3 Программа вычисления переменных затрат (начало)



```

        ' По связям найдем термин "Стоимость возвратных средств"
        For i = 0 To DummyCxns.Count()-1
            Set DummyObjs = DummyCxns.Get(i).TargetObjOcc()
            If DummyObjs.ObjDef().Type = "Technical term" Then
                ' В VC заносится стоимость возвратных средств (со знаком минус)
                VC = -
Val(DummyObjs.ObjDef().Attribute(AT_DESC,SelectedLanguage).GetValue(True))
                CCheck = True
            Else
                ' Если же объект - не тех. термин, то заносим его в CIList (тогда это
кластер)
                CIList.Add(DummyObjs)
            End If
        Next i
        If Not(CCheck) Then
            Incorrect = True ' Если что-то было неправильно, то устанавливаем признак
неверности модели
        End If
    Else
        Incorrect = True
    End If
Else
    Incorrect = True
End If
End Sub

Sub Calc (ByRef CIList As Object, ByRef VC As Double, ByRef Incorrect As Boolean)
    Dim i As Long : Dim j As Long
    Dim CxnsCheck As Long ' Временная переменная для хранения количества связей
    Dim CheckName As String ' Временная переменная для проверки названия объекта
    Dim TempCount As Long ' Счетчик издержек
    Dim Price As Long : Dim Qu As Long ' Для хранения цены и количества
    Dim DummyCxns As Object ' Временная переменная для хранения списка связей
    Dim DummyObj As Object ' Временная переменная для хранения объекта
    Set DummyCxns = CreateObject("ARIS.CxnOccList.5.0"): Set DummyObj =
CreateObject("ARIS.ObjOcc.5.0")
    ' CheckName используется для хранения названия нулевого элемента
    CheckName = CIList.Get(0).ObjDef().Attribute(AT_NAME,SelectedLanguage).GetValue(True)
    For i = 1 To CIList.Count()-1 ' Перебор всех кластеров
        Set DummyCxns = CIList.Get(i).Cxns(EDGES_IN,EDGES_ALL)
        ' Проверка количества связей
        CxnsCheck = Val(CIList.Get(i).ObjDef().Attribute(AT_REM,SelectedLanguage).GetValue(True))
        If CxnsCheck <> DummyCxns.Count()-1 Then
            Incorrect = True
        Else
            ' Если все правильно, то подсчитываем сумму издержек
            TempCount = 0
            ' Перебор всех объектов, связанных с кластером
            For j = 0 To DummyCxns.Count()-1
                Set DummyObj = DummyCxns.Get(j).SourceObjOcc()
                ' Объект "Переменные затраты на один продукт" рассматривать не нужно
                If CheckName <> DummyObj.ObjDef().Attribute(AT_NAME,SelectedLanguage).GetValue(True)
                Then
                    ' Для всех остальных цена умножается на кол-во, и суммируется в
TempCount
                    Price =
Val(DummyObj.ObjDef().Attribute(AT_DESC,SelectedLanguage).GetValue(True))
                    Qu =
Val(DummyObj.ObjDef().Attribute(AT_REM,SelectedLanguage).GetValue(True))
                    TempCount = TempCount + (Price*Qu)
                End If
            Next j
            ' Вычисленная величина заносится в кластер и прибавляется к VC
            CIList.Get(i).ObjDef().Attribute(AT_DESC,SelectedLanguage).SetValue(TempCount,-1)
            VC = VC + TempCount
        End If
    Next i
End Sub

```

Рис.7.5.3 Программа вычисления переменных затрат (окончание)

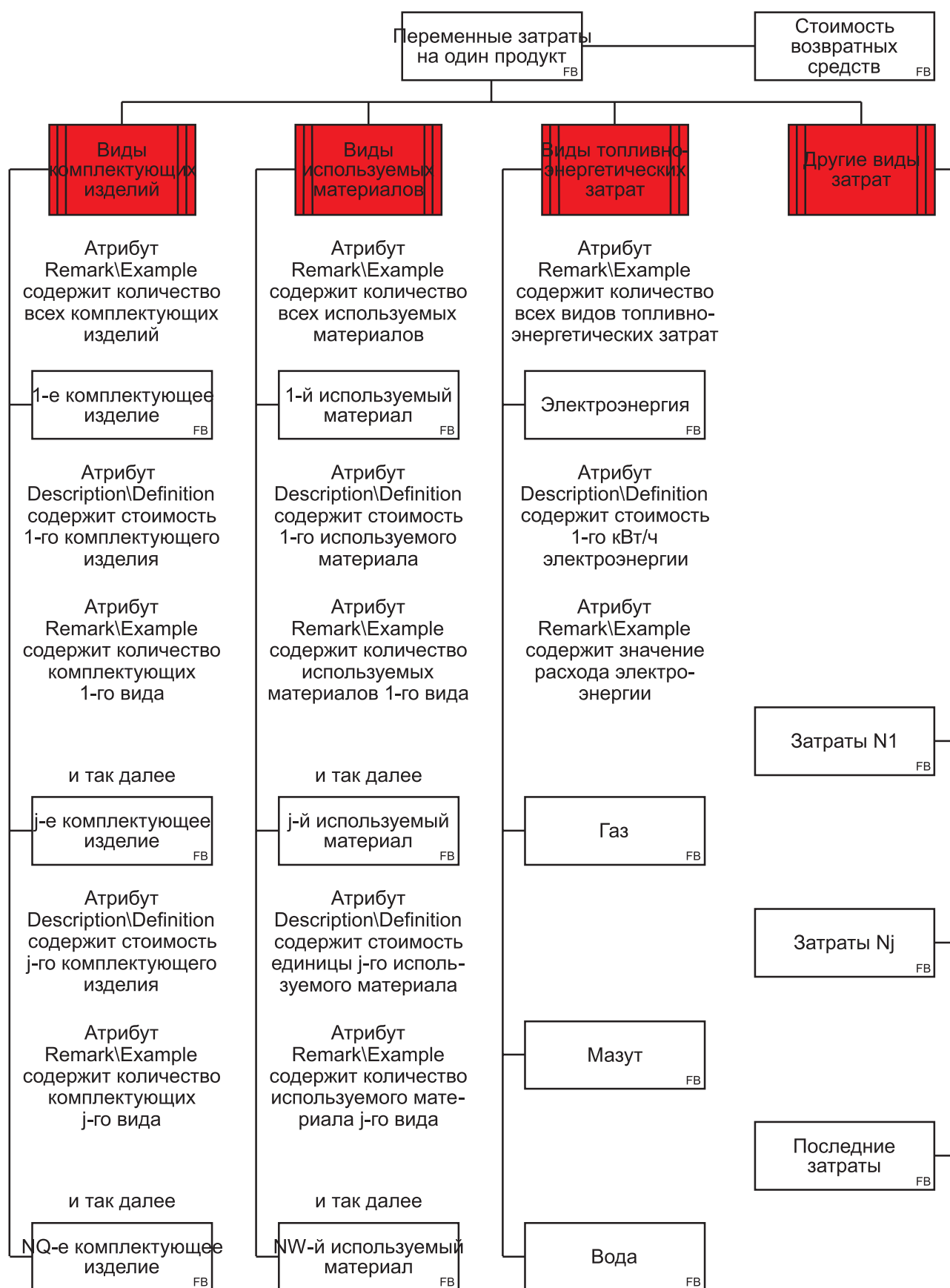


Рис.7.5.4 Структура модели, содержащей входные и выходные данные программы на рис.7.5.3

```

Public OutFile As Object ' Объект для вывода результатов в файл
Sub Main
    Dim i As Long : Dim Models As Object ' Models хранит выбранную модель
    Dim HLFunc As Object ' HLFunc хранит список функций
    Dim LLFunc As Object ' LLFunc хранит список функций, связанных с функциями из списка HLFunc
    Dim NoFunc As Boolean ' Признак окончания работы (когда все функции рассмотрены)
    Set Models = SelectedModels ' Выбор модели
    Set OutFile = CreateObject("ARIS.Output.5.0") : OutFile.Init(OUTTEXT,SelectedLanguage)
    Set HLFunc = CreateObject("ARIS.ObjOccList.5.0") ' Создание списка объектов (функций)
    FirstStep Models,HLFunc ' Выбор и размещение всех функции первого уровня в HLFunc
    While Not(NoFunc)
        Set LLFunc = CreateObject("ARIS.ObjOccList.5.0")
            ' Создание и разрушение списков (HLFunc и LLFunc) в цикле используется для очистки списка
        OneStep HLFunc,LLFunc,NoFunc
        Set HLFunc = Nothing : Set HLFunc = CreateObject("ARIS.ObjOccList.5.0")
        For i = 0 To LLFunc.Count()-1
            HLFunc.Add(LLFunc.Get(i)) ' Переписывает LLFunc в HLFunc
        Next i
        Set LLFunc = Nothing ' Очистка LLFunc
    Wend
    OutFile.WriteReport(SelectedPath,SelectedFile)
End Sub
Sub FirstStep (Models As Object, ByRef HLFunc As Object)
    Dim DummyObjOccs As Object : Dim Check As Long : Dim i As Long
    Set DummyObjOccs = CreateObject("ARIS.ObjOccList.5.0")
    ' Выбирает все функции (OT_FUNC) из модели
    Set DummyObjOccs = Models.Get(0).ObjOccListFilter("",SelectedLanguage,OT_FUNC)
    For i = 0 To DummyObjOccs.Count()-1
        Check = DummyObjOccs.Get(i).InDegree(EDGES_ALL) ' Количество входящих связей
        If Check = 0 Then ' Если входящих связей нет, то функция помещается в HLFunc
            HLFunc.Add(DummyObjOccs.Get(i))
        End If
    Next i
End Sub
Sub OneStep (ByRef HLFunc As Object, ByRef LLFunc As Object, ByRef NoFunc As Boolean)
    Dim i As Long : Dim j As Long : Dim CurrCxnList As Object
    Dim AOC As Currency ' Значение средних прочих издержек
    Dim AOCNew As Currency ' Новое значение средних прочих затрат
    Dim Perc As Long ' Процент вклада
    NoFunc = True : Set CurrCxnList = CreateObject("ARIS.CxnOccList.5.0") ' Создание списка связей
    For i = 0 To HLFunc.Count()-1 ' Перебор всех функций из HLFunc
        ' Получаем значение издержек
        AOC =
        HLFunc.Get(i).ObjDef().Attribute(AT_COST_AVG_OTH,SelectedLanguage).MeasureValue(True)
        Set CurrCxnList = HLFunc.Get(i).Cxls(EDGES_OUT,EDGES_ALL) ' Определяем исходящие
        связи
        If CurrCxnList.Count > 0 Then
            For j = 0 To CurrCxnList.Count()-1
                LLFunc.Add(CurrCxnList.Get(j).TargetObjOcc(),j) ' По связям находим
                функции
                ' Определяем значение процента для каждой функции
                Perc =
                LLFunc.Get(j).ObjDef().Attribute(AT_COST_AVG_PERS,SelectedLanguage).MeasureValue(True)
                AOCNew = Perc/100*AOC ' Вычисляем новое значение
                LLFunc.Get(j).ObjDef().Attribute(AT_COST_AVG_OTH,SelectedLanguage).SetValue(AOCNew,-1)
            Next j
            NoFunc = False
        ' Если была обработана хотя бы одна функция, то снимаем признак окончания работы
        End If
    Next i
End Sub

```

Рис.7.6.1 Программа анализа модели типа Function Tree

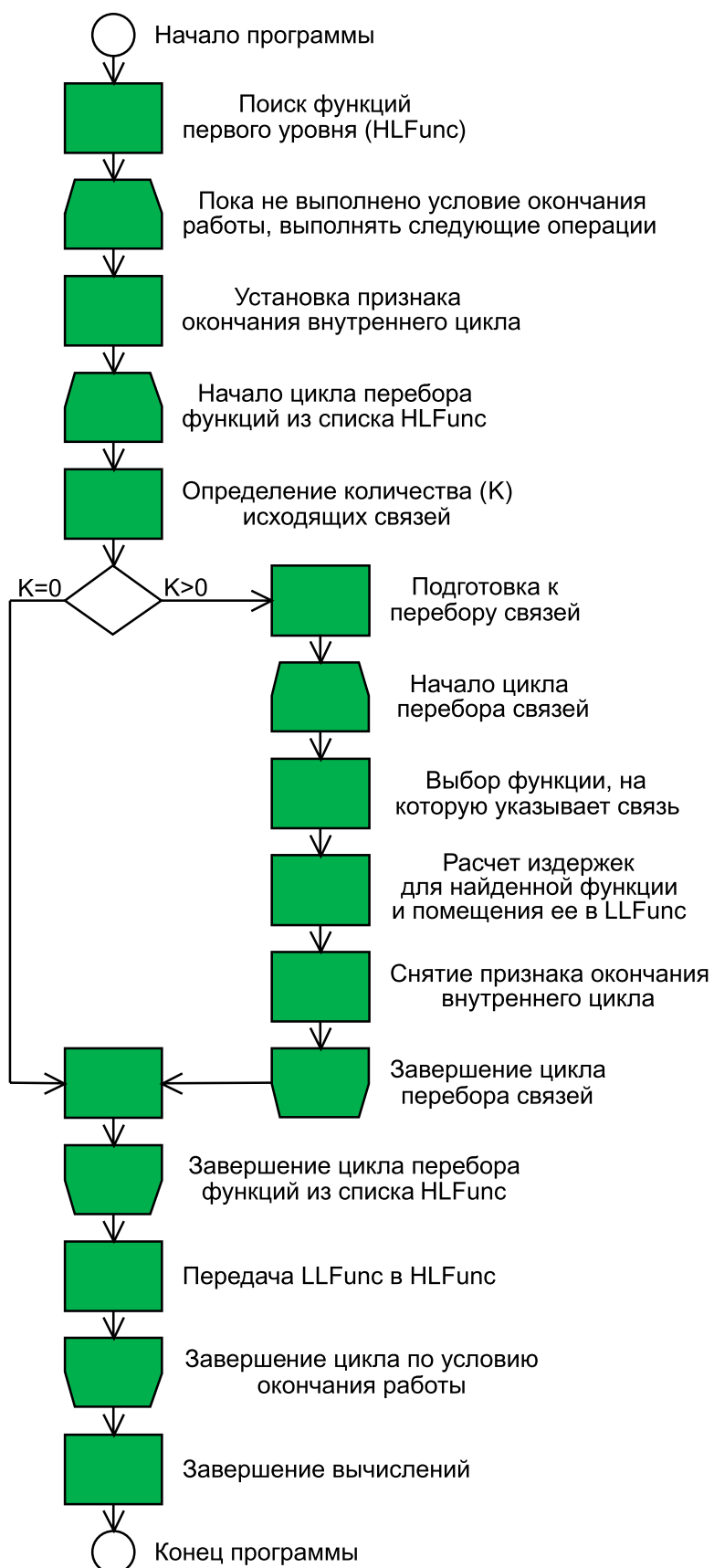


Рис.7.6.2 Блок-схема программы на рис.7.6.1

формирования списка выбранных сотрудников (объект Person) и вывода на печать некоторых анкетных данных и фотографии, привязанной через атрибут Link, по каждому из этих сотрудников. Для работы программы необходимо выбрать модель и формат файла вывода (документ Word). Модель должна содержать объекты типа Person (сотрудник). Заполнять атрибуты анкетных данных (телефон, адрес и т. д.) необязательно. Однако, если они пусты, то в отчет также будут выведены пустые строки.

В ходе работы рассматриваемой программы формируется документ, содержащий данные по сотрудникам в следующем виде.

<Фамилия, имя, отчество>

Адрес: <Адрес>

Телефон: <Номер телефона>

<Фотография>

e-Mail: <Адрес электронной почты>

Должность: <Название должности>

Эта программа использована в последней главе для вывода информации по четырем сотрудникам фирмы, выпускающей мужские сорочки.

### **7.8. Программа “прокрутки” eEPC-модели**

Текст программы последнего учебного скрипта для работы с произвольной eEPC-моделью изображен на

рис.7.8.1. Эта программа отслеживает логику в цепочках выполняемых функций и проходит по объектам выбранных цепочек. Данная программа может быть использована как шаблон (основа) для создания программ имитации и анализа БП.

Для работы программы необходимо в выбранной eEPC-модели БП задать идентификаторы, используемые при обработке логических операторов. В атрибут Remark\Example функций или событий, предшествующих логическому оператору, необходимо внести список идентификаторов выбираемых ветвей (для оператора OR) либо указать один идентификатор (для XOR). Список идентификаторов разделяется пробелами (например, “А Б В Г”, “Вверх Вниз Вправо Влево”). В атрибут Description\Definition каждой функции или события, стоящих после логического оператора, необходимо записать идентификатор выбора ветви. Если в определенном выше списке идентификаторов встретилось совпадение с идентификатором выбора ветви, то данная ветвь считается выбранной, т. е. по ней состоится проход. Например, для списка “А Б В Г”, записанного в объект, предшествующий логическому оператору OR, и идентификаторов выбора ветвей “А”, “Нет”, “Да”, “Б”, записанных в объекты, следующие за этим логическим оператором, будут выбраны ветви “А” и “Б”.

Программа на рис.7.8.1 использовалась для “прокрутки” и выполнения необходимых расчетов в некоторых БП, рассмотренных в последней главе.

```

Public OutFile As Object ' Объект для вывода результатов в файл
Dim Persons As Object ' Список объектов Person
Dim AddArray() As String ' Массивы для диалогового окна
Dim AddedArray() As String
Dim FirstAdded As Boolean ' Признак добавления хотя бы одного элемента в диалог

Sub Main
    Dim Models As Object ' Хранит выбранную модель
    Dim DlgRes As Boolean ' Признак окончания диалога
    Dim SelPers As Object ' Список выбранных персон
    Dim i As Long
    Dim j As Long
    Dim DummyStr As String ' Рабочая (временная) строка
    Set OutFile = CreateObject("ARIS.Output.5.0") ' Создание и инициализация файла отчета
    OutFile.Init(OUTWORD,SelectedLanguage)
    Set Models = SelectedModels ' Определяем выбранную модель
    Set Persons = CreateObject("ARIS.ObjOccList.5.0") ' Создание списков
    Set SelPers = CreateObject("ARIS.ObjOccList.5.0")
    FindPersons Models,Persons ' Выделяет все объекты Persons из модели
    DlgRes = PersSelectDlg ' Запускает диалог
    If Not(FirstAdded) Then ' Если в список помещен хоть один объект
        ' выбираем нужные объекты из Persons и помещаем их в SelPers
        For i = 0 To Persons.Count()-1
            DummyStr = Persons.Get(i).ObjDef().Attribute(AT_NAME,SelectedLanguage).GetValue(True)
            For j = 0 To UBound(AddedArray)
                If DummyStr = AddedArray(j) Then
                    SelPers.Add(Persons.Get(i))
                End If
            Next j
        Next i
        Report SelPers ' По выбранным персонам создается отчет
    End If
End Sub

Sub FindPersons (Models As Object, ByRef Persons As Object)
    Dim DummyObjs As Object
    Dim i As Long
    ' Типовая подпрограмма по выбору объектов Persons из модели
    Set DummyObjs = CreateObject("ARIS.ObjOccList.5.0")
    Set DummyObjs = Models.Get(0).ObjOccListFilter("",SelectedLanguage,OT_PERS)
    If DummyObjs.Count() > 0 Then
        For i = 0 To DummyObjs.Count()-1
            Persons.Add(DummyObjs.Get(i))
        Next i
    End If
End Sub

Function PersSelectDlg As Boolean ' Подпрограмма диалога
    Begin Dialog UserDialog 530,252,.DialogFunc ' Создано с помощью утилиты Dialog Editor
        ListBox 20,14,490,84,AddArray(),.AddList
        ListBox 20,133,490,84,AddedArray(),.RmList
        PushButton 130,105,100,21,"Добавить >>",>.PushAdd
        PushButton 300,105,100,21,"Удалить <<",>.PushRemove
        OKButton 410,224,90,21,.OK
    End Dialog
    Dim dlg As UserDialog
    PersSelectDlg = Dialog(dlg) ' Запуск диалога
End Function

```

Рис.7.7.1 Программа анализа организационной диаграммы (начало)



```

Sub Report(SelPers As Object)
    Dim i As Long : Dim j As Long
    Dim Photo As Object ' Для хранения фотографий
    Dim Names() As String ' Массивы для хранения анкетных данных
    Dim Tel() As String : Dim Addr() As String : Dim eMail() As String : Dim Position() As String
    Dim DummyCxns As Object : Dim StrData As String
    ReDim Names(0 To SelPers.Count()-1) : ReDim Tel(0 To SelPers.Count()-1)
    ReDim Addr(0 To SelPers.Count()-1) : ReDim eMail(0 To SelPers.Count()-1)
    ReDim Position(0 To SelPers.Count()-1) : Set Photo = CreateObject("ARIS.Picture.5.0")
    Set DummyCxns = CreateObject("ARIS.CxnOccList.5.0")
    OutFile.DefineF("A","Times New Roman",12,C_BLACK,COLOR_TRANSPARENT,FMT_LEFT,0,0,0,0,0)
    OutFile.DefineF("B","Times New Roman",12,C_BLACK,COLOR_TRANSPARENT,FMT_ITALIC Or
    FMT_LEFT,0,0,0,0,0)
    For i = 0 To SelPers.Count()-1 ' Перебор анкетных данных

Names(i) = SelPers.Get(i).ObjDef().Attribute(AT_NAME,SelectedLanguage).GetValue(True)

Tel(i) = SelPers.Get(i).ObjDef().Attribute(AT_PHONE_NUM,SelectedLanguage).GetValue(True)
Addr(i) = SelPers.Get(i).ObjDef().Attribute(AT_ADDR,SelectedLanguage).GetValue(True)
eMail(i) = SelPers.Get(i).ObjDef().Attribute(AT_EMAIL_ADDR,SelectedLanguage).GetValue(True)
' Определение должности (через связь с объектом Position)
Set DummyCxns = SelPers.Get(i).Cxns(EDGES_OUT,EDGES_ALL)
For j = 0 To DummyCxns.Count()-1
    If DummyCxns.Get(j).TargetObjOcc().ObjDef().Type() = "Position" Then
        Position(i) =
        DummyCxns.Get(j).TargetObjOcc().ObjDef().Attribute(AT_NAME,SelectedLanguage).GetValue(True)
    End If
Next j
Next i
' Вывод отчета
OutFile.TopMargin = 15 : OutFile.LeftMargin = 15 : OutFile.RightMargin = 15 : OutFile.BottomMargin = 15
' Начало таблицы
OutFile.BeginTable(100,COLOR_TRANSPARENT,COLOR_TRANSPARENT,FMT_LEFT,0)
Const LF = Chr(13) + Chr(10) ' Константа перевода строки (Enter)
For i = 0 To SelPers.Count()-1
    OutFile.TableRow() : StrData = Names(i) : OutFile.TableCellF(StrData,100,"B") : OutFile.TableRow()
    ' Все данные выводятся в одной ячейке таблицы
    StrData = LF+"Адрес : "+Addr(i)+LF+"Телефон : "+Tel(i)+LF+"e-Mail : "+eMail(i)+LF+"Должность :
    "+Position(i)+LF
    OutFile.TableCellF(StrData,100,"A")
Next i
OutFile.EndTable("",100,"Times New Roman",12,C_BLACK,COLOR_TRANSPARENT,0,FMT_LEFT,0)
OutFile.WriteReport(SelectedPath,SelectedFile) ' Вывод отчета
End Sub

Dim Add As Boolean ' Признак добавления элемента
Dim Remove As Boolean ' Признак удаления элемента
Dim AddPos As Integer ' Позиция добавляемого элемента
Dim RemovePos As Integer ' Позиция удаляемого элемента

Function DialogFunc%(DlgItem$, Action%, SuppValue%) ' Подпрограмма обработки событий диалога
    Dim i As Long
    Dim DummyStr As String
    ' Вид события определяется переменной Action%
    Select Case Action%
        Case 1 ' Инициализация диалога
            ReDim Preserve AddArray(0 To Persons.Count()-1)
            For i = 0 To Persons.Count()-1
                AddArray(i) =
Persons.Get(i).ObjDef().Attribute(AT_NAME,SelectedLanguage).GetValue(True)
            Next i
            FirstAdded = True

```

Рис.7.7.1 Программа анализа организационной диаграммы (продолжение)

```

Case 2 ‘ Изменение состояния списков (при выборе элемента из списка)
  Select Case DlgItem$ ‘ Имя элемента, вызвавшего событие, хранится в DlgItem$
    Case “AddList”
      ‘ При выборе элемента устанавливается признак добавления и определяется
позиция элемента
      If AddArray(SuppValue) <> “” Then
        Add = True : AddPos = SuppValue
      End If
      Remove = False
      ‘ При выборе элемента из итогового списка устанавливается признак удаления и
позиция элемента
      Case “RmList”
        If AddedArray(SuppValue) <> “” Then
          Remove = True : RemovePos = SuppValue
        End If
        Add = False
        ‘ При нажатии на кнопку “Добавить” (в случае установленного признака Add)
        ‘ происходит удаление элемента из исходного списка и добавление в итоговый
      Case “PushAdd”
        If Add Then
          ‘ Подпрограммы добавления и удаления работают одинаково
          ‘ элемент удаляется из одного списка, и добавляется в другой
          DummyStr = AddArray(AddPos) : AddArray(AddPos) =
AddArray(UBound(AddArray))
          If UBound(AddArray) = 0 Then
            ReDim AddArray (0)
          Else
            ReDim Preserve AddArray(0 To (UBound(AddArray)-1))
          End If
          If FirstAdded Then
            ReDim AddedArray(0) : FirstAdded = False
          Else
            ReDim Preserve AddedArray(0 To (UBound(AddedArray)+1))
          End If
          AddedArray(UBound(AddedArray)) = DummyStr
          DlgListBoxArray “AddList”,AddArray() : DlgListBoxArray “RmList”,AddedArray()
        End If
        ‘ При нажатии на кнопку “Удалить” (в случае установленного признака Remove)
        ‘ происходит удаление элемента из итогового списка и добавление в исходный
      Case “PushRemove”
        If Remove Then
          DummyStr = AddedArray(RemovePos)
          AddedArray(RemovePos) = AddedArray(UBound(AddedArray))
          If UBound(AddedArray) = 0 Then
            ReDim AddedArray (0)
          Else
            ReDim Preserve AddedArray(0 To (UBound(AddedArray)-1))
          End If
          ReDim Preserve AddArray(0 To (UBound(AddArray)+1))
          AddArray(UBound(AddArray)) = DummyStr
          DlgListBoxArray “AddList”,AddArray() : DlgListBoxArray “RmList”,AddedArray()
        End If
      End Select
      ‘ Если нажата кнопка “ОК”, то закрыть диалог
      If DlgItem$ = “OK” Then
        DialogFunc% = False
      Else
        DialogFunc% = True
      End If
      ‘ Остальные случаи в данном диалоге не используются
    Case 3 : Case 4 : Case 5 : Case 6
  End Select
End Function

```

Рис.7.7.1 Программа анализа организационной диаграммы (окончание)

```

Public OutFile As Object ' Объект для вывода результатов в файл
Dim EOS As Boolean ' Признак окончания анализа процесса
Dim First As Boolean ' Признак первого элемента цепочки
Dim Deleted As Long ' Количество завершенных цепочек

Sub Main
    Dim Models As Object ' Хранит выбранную модель
    Dim Start As Object ' Хранит начальные элементы цепочки
    Dim SimList() As String ' Список, выводимый в отчет
    Set OutFile = CreateObject("ARIS.Output.5.0")
    OutFile.Init(OUTTEXT,SelectedLanguage)
    Set Models = SelectedModels ' Определяем выбранную модель
    Set Start = CreateObject("ARIS.ObjOccList.5.0") ' Создание списка
    FindStart Models,Start ' Находим первый элемент цепочки процесса
    First = True : Deleted = 1 ' Установка признака и переменной Deleted
    While Not(EOS)
        ' Выполнять единичный "шаг" до тех пор, пока не будет установлен признак окончания
        MakeStep Start,SimList()
    Wend
    Report SimList() ' Формирует отчет
End Sub

Sub Report (SimList() As String)
    Dim i As Long
    OutFile.DefineF("1","Arial",10,C_BLACK,COLOR_TRANSPARENT,FMT_LEFT,0,0,0,0,0)
    OutFile.WriteReport(SelectedPath,SelectedFile)
End Sub

Sub FindStart (Models As Object, ByRef Start As Object)
    Dim i As Long
    Dim DummyObjs As Object
    Dim DummyCxns As Object
    Set DummyObjs = CreateObject("ARIS.ObjOccList.5.0")
    Set DummyObjs = Models.Get(0).ObjOccList ' Выбираем все объекты из модели
    Set DummyCxns = CreateObject("ARIS.CxnOccList.5.0")
    ' Выбираем из всех объектов те, которые служат началом цепочки
    For i = 0 To DummyObjs.Count()-1
        ' У всех начальных элементов отсутствует входная связь
        Set DummyCxns = DummyObjs.Get(i).Cxns(EDGES_IN,EDGES_ALL)
        If DummyCxns.Count() = 0 Then
            Start.Add(DummyObjs.Get(i))
        End If
    Next i
End Sub

Sub MakeStep (ByRef Start As Object, ByRef SimList() As String)
    Dim i As Long
    Dim DummyCxns As Object
    Dim DummyObj As Object
    Set DummyCxns = CreateObject("ARIS.CxnOccList.5.0")
    Set DummyObj = CreateObject("ARIS.ObjOcc.5.0")

```

Рис.7.8.1 Программа анализа eEPC-диаграмм (начало)

```

If Start.Count() = 0 Then
    ' Если в списке Start не осталось ни одного элемента, то завершить работу
    EOS = True
Else ' В противном случае выполняется движение на один элемент по всем цепочкам
    ' т.к. во время цикла число элементов в Start изменяется, то используется
While, а не For
    i = 0
    While i <= Start.Count()-1
        ' Определяет все связи, исходящие из элемента Start
        Set DummyCxns = Start.Get(i).Cxns(EDGES_OUT,EDGES_STRUCTURE)
        If First Then ' Если установлен признак первого элемента, то
            End If
        If DummyCxns.Count() <> 0 Then ' Если существует хотя бы одна связь, то
            ' Определяет объект по этой связи
            Set DummyObj = DummyCxns.Get(0).TargetObjOcc()
            ' Логические операторы обрабатываются отдельно
            If DummyObj.ObjDef().Type() = "Rule" Then
                ' Для них используется подпрограмма выбора путей
                SelectPath DummyObj,Start,i,SimList()
                First = True
            ' Т.к. в процессе работы SelectPath обрабатываемый элемент удаляется, то i уменьшается
            i = i - 1
            Else ' Если встретился не лог. оператор, то происходит передвижение
            ' Элемент из Start удаляется, а вместо него добавляется следующий (по цепочке)
            Start.Delete(i)
            Start.Add(DummyObj,i)
            End If
        Else ' Если выходящих связей нет, то элемент удаляется из Start
            Start.Delete(i)
            ' При удалении элемента i необходимо уменьшить
            i = i - 1
            ' Для сохранения правильной нумерации Deleted увеличивается
            Deleted = Deleted + 1
        End If
        ' При следующем проходе цикла будет обработан следующий элемент
        i = i + 1
    Wend
    If First Then First = False ' В конце шага признак First снимается
End If
End Sub

Sub SelectPath (Opr As Object,ByRef Start As Object, Pos As Long,ByRef SimList() As String)
    Dim i As Long
    Dim j As Long
    Dim DummyCxns As Object
    Dim OprName As String
    Dim TmpStr As String
    Dim TmpStr1 As String
    Dim TmpChar As String
    Dim IDs() As String
    Dim IDCounter As Long
    Dim Added As Long
    Dim Selected As Boolean
    Set DummyCxns = CreateObject("ARIS.CxnOccList.5.0")
    Set DummyCxns = Opr.Cxns(EDGES_OUT,EDGES_STRUCTURE)
    OprName = Opr.ObjDef().Name(SelectedLanguage) ' Получение имени оператора
    ' начало выбора идентификаторов
    TmpStr = Start.Get(Pos).ObjDef().Attribute(AT_REM,SelectedLanguage).GetValue(True)
    TmpStr1 = "" : IDCounter = 0 : ReDim IDs(0)

```

Рис.7.8.1 Программа анализа eEPC-диаграмм (продолжение)

```

For i = 1 To Len(TmpStr)
    TmpChar = Mid(TmpStr,i,1)
    If TmpChar = " " Or i = Len(TmpStr) Then
        ReDim Preserve IDs(0 To IDCounter)
        If i = Len(TmpStr) Then
            IDs(IDCounter) = TmpChar
        Else
            IDs(IDCounter) = TmpStr1
        End If
        IDCounter = IDCounter + 1
        TmpStr1 = ""
    Else
        TmpStr1 = TmpStr1 + TmpChar
    End If
Next i
Start.Delete(Pos)
'конец выбора идентификаторов
' Данный блок используется для выбора элементов, разделенных пробелами, из строки
' и заполнения их в массив (IDs)
' т.е. строка "А Б В" распадется на три элемента - в IDs(0) будет "А", а в IDs(2) - "В"
Select Case OprName ' Выбор из трех операторов: AND, OR, XOR
Case "AND operator" ' В случае оператора AND выбираются все ветви
    For i = 0 To DummyCxns.Count()-1
        Start.Add(DummyCxns.Get(i).TargetObjOcc())
    Next i
Case "OR operator"
    Added = 0 ' Added используется для подсчета количества выбранных ветвей
    Selected = False
    For i = 0 To DummyCxns.Count()-1
        ' Сравниваем идентификаторы ветвей с элементами IDs (см. выше) для выбора дальнейшего пути
        TmpStr = DummyCxns.Get(i).TargetObjOcc.ObjDef().Attribute(AT_DESC,SelectedLanguage).GetValue(True) =
        For j = 0 To UBound(IDs)
            If TmpStr = IDs(j) Then
                Selected = True ' Если выбор произошел, то установить признак
                Start.Add(DummyCxns.Get(i).TargetObjOcc)
                Added = Added + 1
            End If
        Next j
    Next i
    If Selected = False Then ' Если ранее не было выбрано ни одной ветви, то выбрать все
        For i = 0 To DummyCxns.Count()-1
            Start.Add(DummyCxns.Get(i).TargetObjOcc())
        Added = Added + 1
    Next i
End If
Case "XOR operator" ' Так же, как OR, но выбирается только одна ветвь
    Selected = False
    For i = 0 To DummyCxns.Count()-1
        TmpStr = DummyCxns.Get(i).TargetObjOcc.ObjDef().Attribute(AT_DESC,SelectedLanguage).GetValue(True)
        For j = 0 To UBound(IDs)
            If TmpStr = IDs(j) And Selected = False Then
                Selected = True
                Start.Add(DummyCxns.Get(i).TargetObjOcc)
            End If
        Next j
    Next i
    If Selected = False Then
        Start.Add(DummyCxns.Get(0).TargetObjOcc())
    End If
End Select
End Sub

```

Рис.7.8.1 Программа анализа eEPC-диаграмм (окончание)

## 8. ПРИМЕРЫ МОДЕЛИРОВАНИЯ СИСТЕМ И ПРОЦЕССОВ

*В этой главе рассмотрены различные примеры систем и их процессов. Для них построены основные ARIS-диаграммы. Мы не стремились привести полное описание рассматриваемых систем и процессов, а также не ставили перед собой конкретные цели такого описания. Наша задача заключалась в демонстрации примеров вербального (словесного) и соответствующего формального (диаграммного) описания рассматриваемых систем и их процессов. Полное описание, например, деятельности научно-производственной фирмы, которое вкратце рассмотрено в последнем разделе этой главы, требует на порядок большего места, оцифровки сотен атрибутов, больших усилий специалистов и, соответственно, дорого стоит.*

### 8.1. Обслуживание клиентов гостиницы

На рис.8.1.1 представлена диаграмма подчиненности субъектов ответственности небольшой гостиницы. Из этой диаграммы видно, сколько человек работает в гостинице, в том числе, сколько охранников, горничных и других служащих, а также какие должности есть в штатном расписании гостиницы и как они подчинены друг другу. Эта диаграмма может быть полезна всем работникам гостиницы, ее клиентам, страховым агентам, налоговой инспекции, кредиторам и т. д.

Обслуживающий персонал гостиницы и клиенты могут узнать, к кому они могут обратиться с просьбой или с жалобой при возникновении конфликта. В свою очередь, новый или потенциальный сотрудник гостиницы сможет увидеть, кто будет ему подчиняться, от кого он сам будет получать распоряжения и т. д. Таким образом, данная диаграмма позволяет определить структуру штата и иерархию подчиненности всех сотрудников гостиницы.

Из диаграммы на рис.8.1.2 видно, что на втором этаже гостиницы находятся номера для проживания постояльцев. Второй этаж обслуживают технички, коридорный и горничные. В одноместном номере есть спальня и ванная и т. д. Эта диаграмма может использоваться в качестве ознакомительного материала для новых работников гостиницы, а также для комиссий из любых проверяющих организаций.

На рис.8.1.3 изображена диаграмма основных процессов управления гостиницей. Опишем подробнее процесс заселения клиента в номер, который протекает по следующему сценарию.

Клиент входит в гостиницу и обращается к портье с просьбой предоставить ему номер. Портье проверяет, забронирован номер или нет. Если номер не забронирован, портье проверяет наличие свободных номеров или мест. Если таковых нет, то он сообщает об этом клиенту и тот покидает гостиницу. Если свободные номера (места) есть,

то портье просит клиента заполнить анкету. В ней указывается имя, дата вселения, предполагаемая длительность проживания и другие данные о клиенте, которые необходимы для его вселения в гостиницу. Клиент заполняет анкету и отдает ее портье, который проверяет правильность заполнения анкеты. Если она заполнена неправильно, то он возвращает ее клиенту с замечаниями и просит заполнить анкету заново должным образом. Если анкета заполнена правильно или номер для клиента забронирован заранее, то портье просит клиента оплатить номер. Клиент идет в кассу и оплачивает проживание в номере. Затем клиент возвращается к портье, который заносит данные о клиенте и об оплате в электронную базу данных, после чего выдает клиенту ключ от номера. Клиент вселен.

eEPC-диаграмма описанного процесса изображена на рис.8.1.4.

### 8.2. Услуги интернет-салона

Рассмотрим процесс предоставления услуг интернет-салона. Опишем его с точки зрения клиента, т. е. потребителя данных услуг. Поэтому описываемый процесс не включает в себя такие вопросы, как получение прибыли, техническое обслуживание оборудования и многое другое.

Началом процесса служит факт появления клиента в офисе интернет-салона, где размещено необходимое оборудование и осуществляется обслуживание клиента. Работу с клиентом проводит оператор. Он готовит оборудование в соответствии с требованиями клиента. Информация о перечне и стоимости услуг, предоставляемых салоном, сосредоточена в *прайс-листе*, из которого клиент, например, узнает, что для него возможна:

- работа на компьютере, подключенном к сети Интернет;
- работа на компьютере в локальной сети (как правило, сетевые игры);
- печать на принтере;
- ксерокопирование;



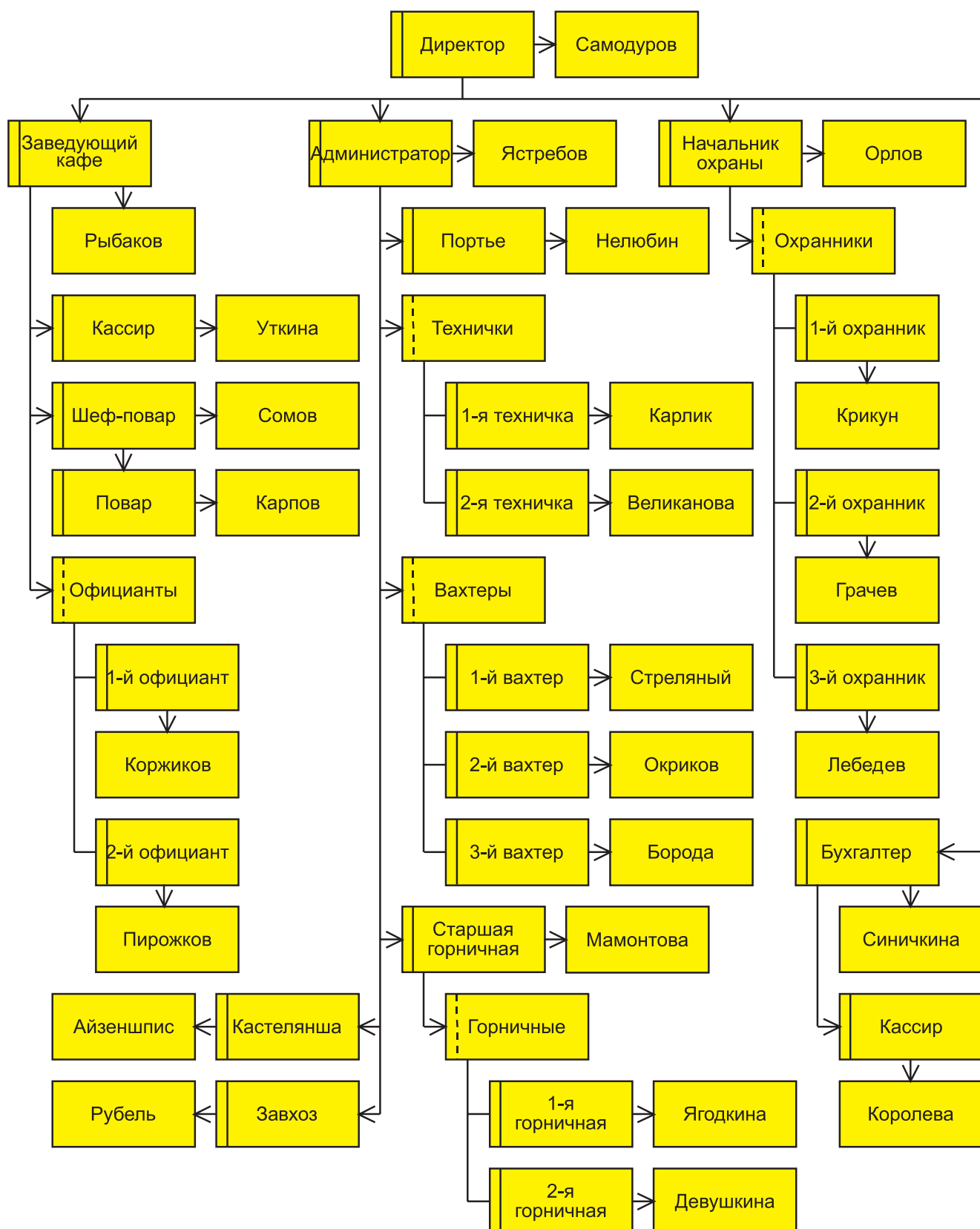


Рис.8.1.1 Диаграмма подчиненности субъектов ответственности гостиницы

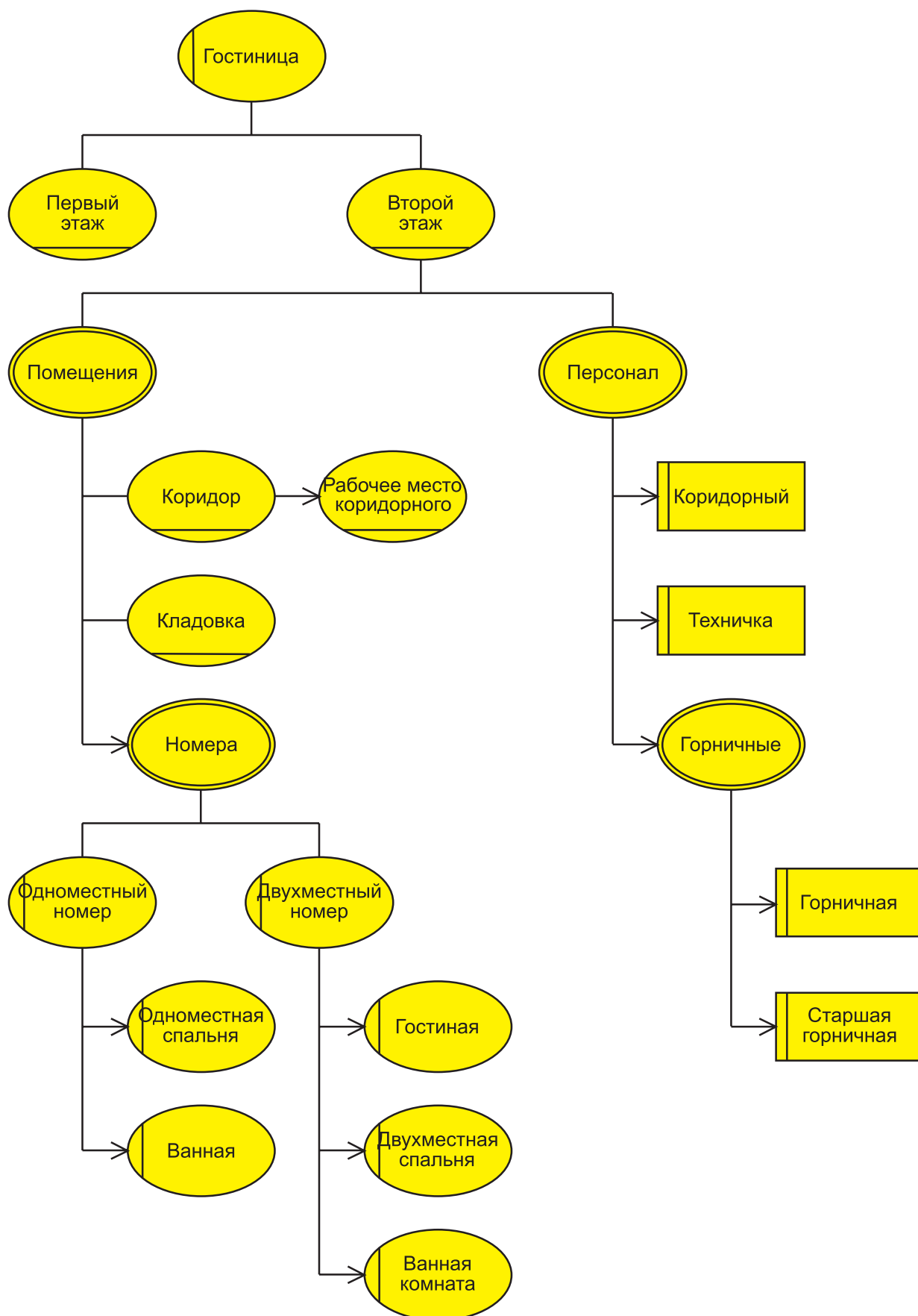


Рис.8.1.2 Диаграмма размещения организационных единиц

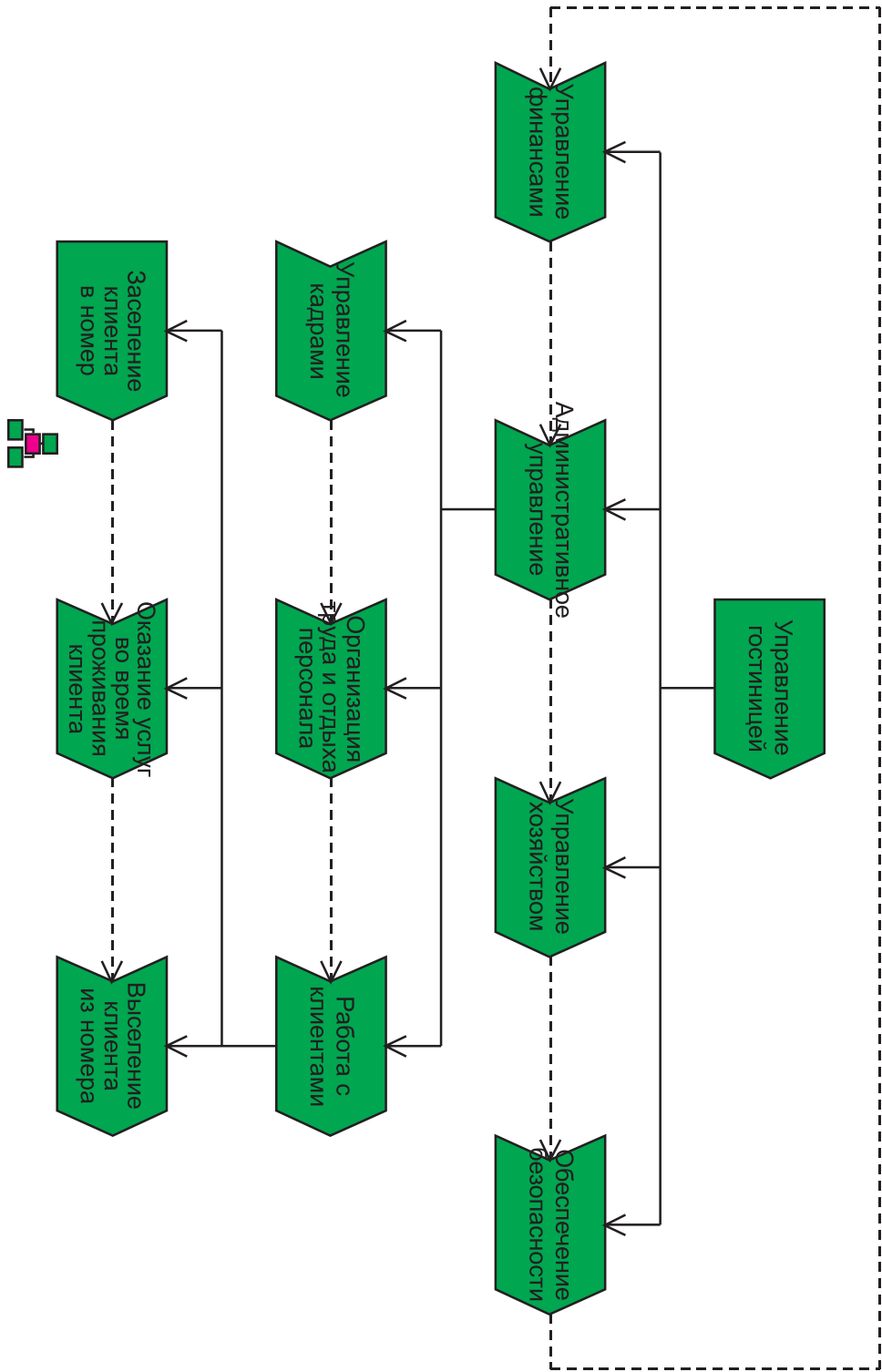


Рис.8.1.3 Диаграмма основных процессов управления гостиницей

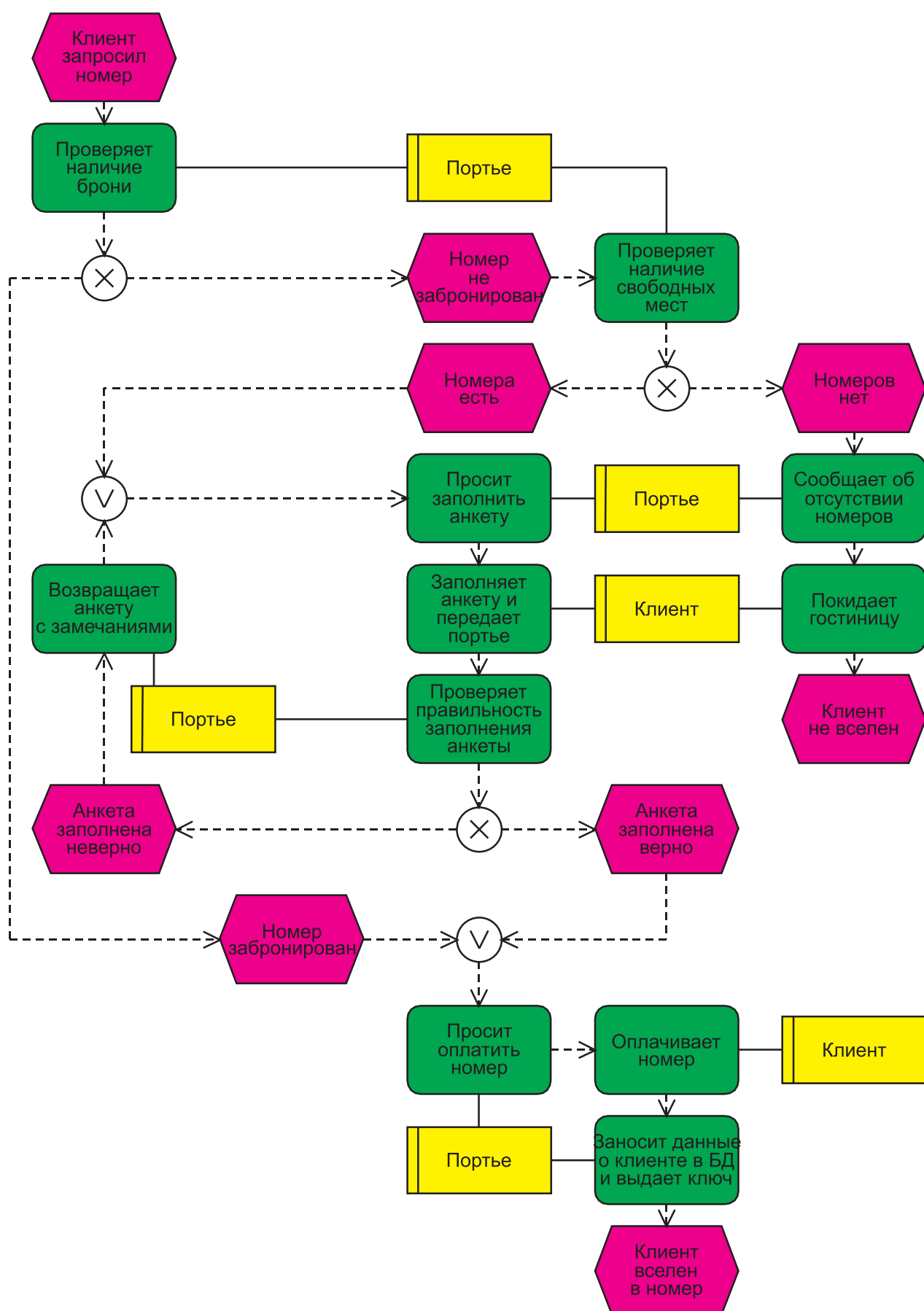


Рис.8.1.4 eEPC-модель заселения клиента в номер

- сканирование и распознавание текста.

Клиент заказывает необходимые ему услуги. Оператор подсчитывает общую стоимость заказа и сообщает ее клиенту. Клиент производит полную оплату заказа либо отказывается от него (например, в случае нехватки денег). В случае оплаты заказа клиент приступает к использованию всех оплаченных услуг. При этом он может обратиться к помощнику оператора с целью получения необходимой информации о работе на компьютере. В случае возникновения каких-либо форс-мажорных обстоятельств (например, отключение электроэнергии) клиент вправе потребовать у оператора деньги за оставшуюся неиспользованной часть заказа.

Рассмотрим подробнее процесс предоставления конкретных услуг. Пусть, например, клиент заказал работу на компьютере, подключенном к сети Интернет, с последующей распечаткой желаемой информации на принтере.

При работе в Интернете клиент оплачивает как объем полученной информации, так и время работы за компьютером (так как это фактически «работа на компьютере в локальной сети» – одна из услуг). Поэтому перед началом работы клиент оплачивает определенное время, которое он планирует провести за компьютером, а также трафик, т. е. определенный объем информации, полученной из Интернета. Если клиент полностью израсходовал трафик, но не израсходовал время, то он может обратиться к оператору с просьбой увеличить трафик. После оплаты дополнительного трафика клиент продолжает свою работу. Если же клиент израсходовал заказанное время, то он может заказать и оплатить дополнительное время только в случае отсутствия очереди, т. е. если другие клиенты не оплатили вперед работу за этим компьютером. В случае окончания времени и отсутствия желания или возможности продолжать работу, оператор обязан вернуть клиенту часть заплаченной им суммы за неиспользованный трафик (если таковой имеется). После окончания работы в Интернете клиент может распечатать необходимую ему информацию на принтере. Для этого он должен указать оператору файл, в котором содержится эта информация. После выяснения количества страниц, которые необходимо распечатать, оператор сообщает клиенту общую стоимость этой услуги. После оплаты этой стоимости оператор производит печать.

На рис.8.2.1, 8.2.2 приведены диаграммы, формально описывающие интернет-салон и основные процессы, в нем происходящие. Рекомендуем читателю самостоятельно построить eEPC-диаграмму на основе сценария, описанного в предшествующем абзаце.

### 8.3. Телевизионные игры

Каждый из нас наблюдал те или иные телевизионные игры, например, «О, счастливики!», «Поле чудес», «Звездный час», «Что? Где? Когда?», «Алчность», «Как стать миллионером», «Своя игра» и др. На основе наблюдений можно построить формальное (диаграммное) описание этих игр, например, с целью их перевода на современную технологическую базу, т. е. реализацию этих игр в виде е-процессов или m-процессов, в которых все участники игры могут находиться в любом месте (дома, в офисе, на берегу моря, в автомобиле и т. д.) и принимать участие в игре или наблюдать за ней, имея для этого при себе только Notebook, мобильные средства связи и соответствующее ПО. В качестве примера построим некоторые из ARIS-моделей игры «Поле чудес».

«Поле чудес» – это познавательная и развлекательная игра, направленная на выявление наиболее достойных и интересных людей в предварительном отборе и знакомство их с телезрителями. Игра состоит из трех отборочных туров, финала и возможной суперигры. После одного из туров возможна игра со зрителями. Каждый из перечисленных процессов состоит из подпроцессов.

Отборочные туры и финал (четвертый тур) проходят по одному и тому же сценарию. Отборочный тур начинают три игрока, и он заканчивается выявлением одного финалиста. В исключительном случае финалист может быть не определен (не выявлен), например, когда всем трем участникам тура выпал сектор «Приз», и они согласились взять приз, тем самым закончив игру. Здесь этот и другие исключительные случаи не рассматриваются, т. е. всегда определены три финалиста и победитель игры.

К рассматриваемой игре имеют отношение следующие субъекты: ведущий, участник, финалист, зритель, группа участников, группа финалистов, группа зрителей, победитель игры (финала), победитель суперигры, ассистенты ведущего. Игра проходит в студии, где расположены места для зрителей, трех игроков и ведущего, а также табло и барабан, который имеет сектора, указатель и рукоятку. На рис.8.3.1 изображена диаграмма рабочих мест в студии. На рис.8.3.2 изображена диаграмма технических терминов (понятий), связанных с понятием «Барабан».

В игре «Поле чудес» можно выделить три комплексных процесса «Игра с игроками», «Игра со зрителями» и «Суперигра». На рис.8.3.3 изображена диаграмма основных процессов и подпроцессов игры «Поле чудес», в которой отображена их иерархическая подчиненность и последовательность выполнения. Из этого рисунка видно, что функции «Игра в туре», «Суперигра», «Игра две шкатулки» имеют детализирующую связь с другими

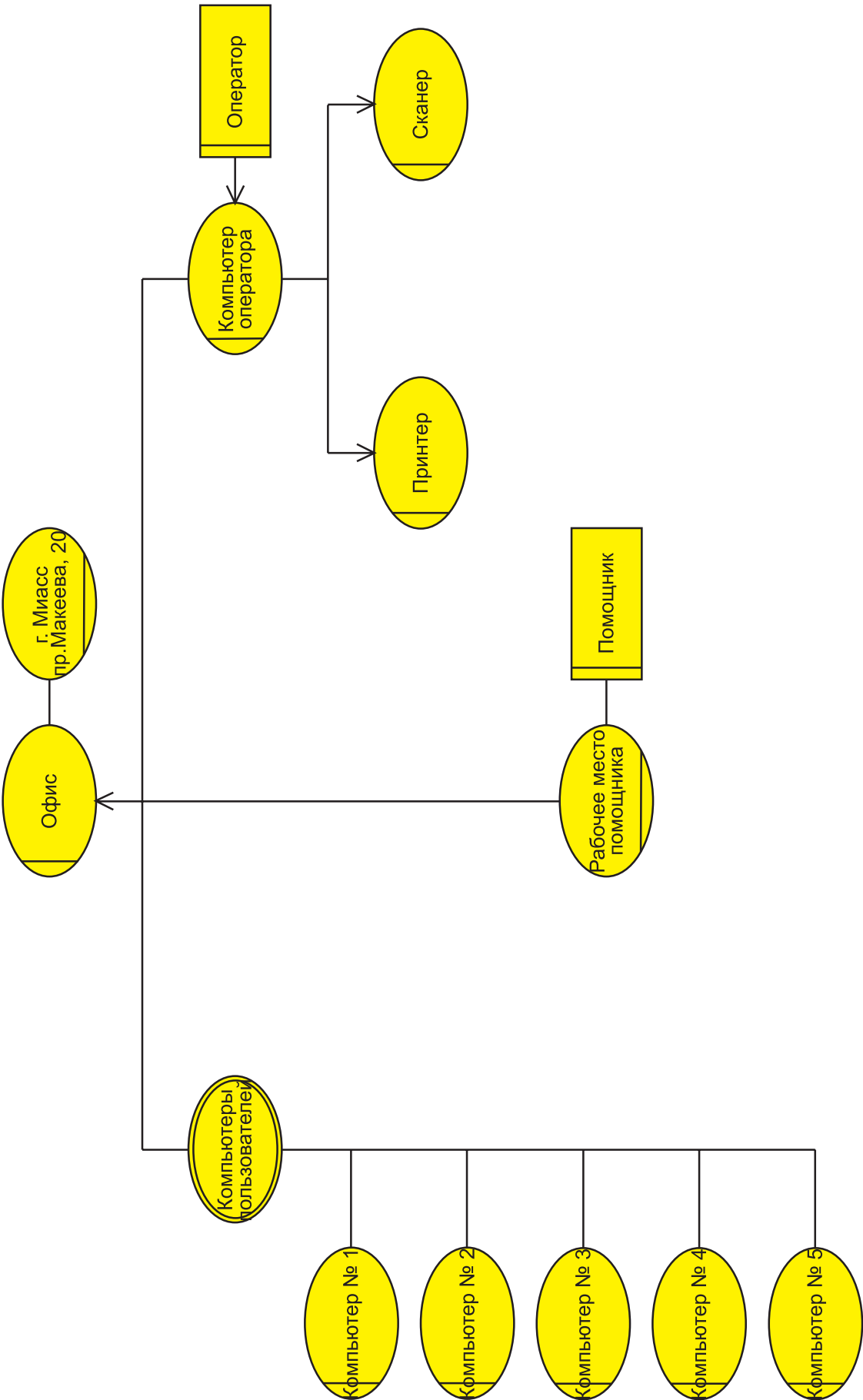


Рис.8.2.1 Организационная диаграмма компьютерного салона



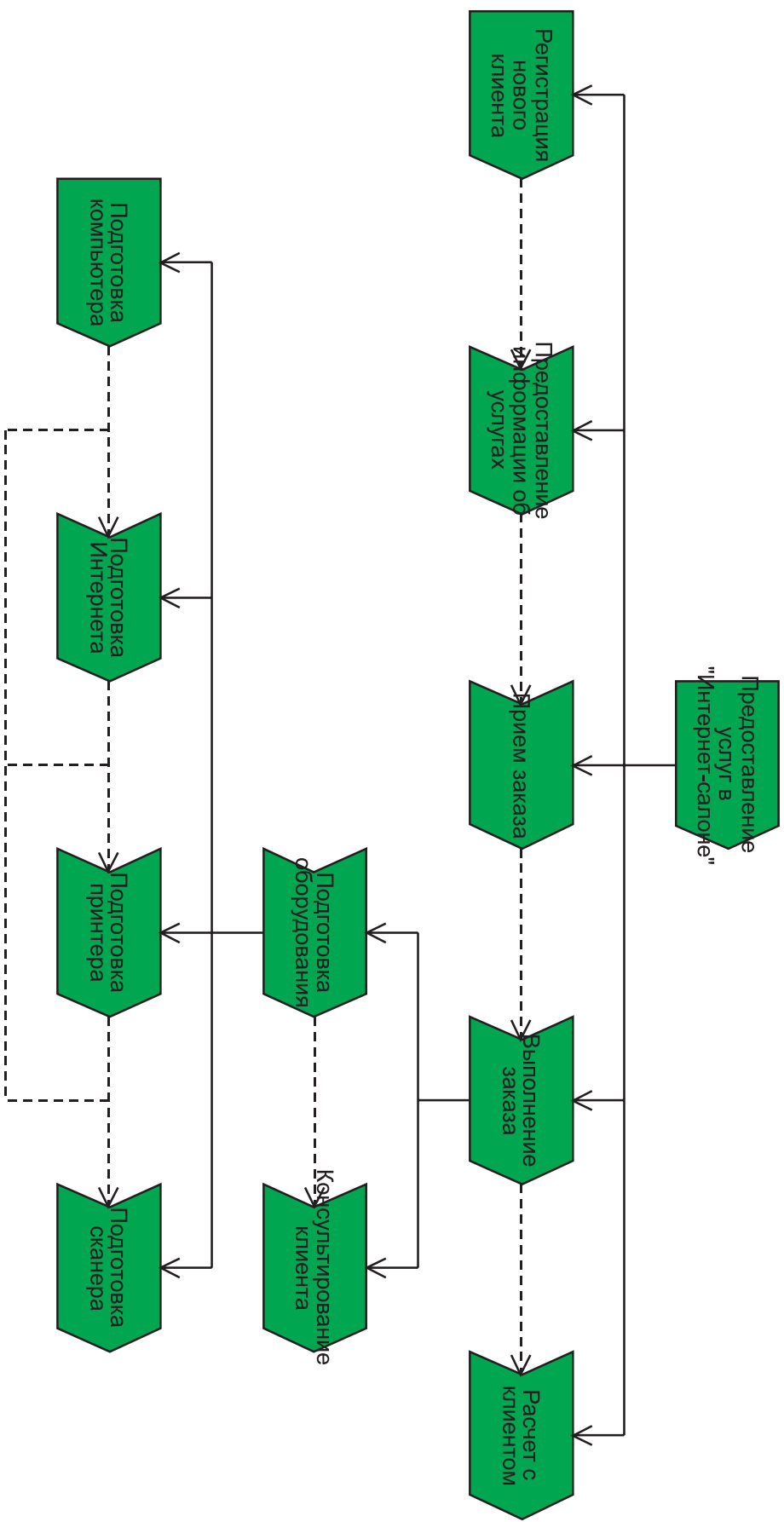


Рис.8.2.2 Диаграмма основных процессов компьютерного салона

диаграммами, которые подробно описывают эти составляющие игры “Поле чудес”.

Начнем с рассмотрения суперигры, в которой участвует победитель финала. Если он соглашается на суперигру, то она протекает по следующему сценарию. Ведущий раскладывает на барабане список призов. Игрок вращает барабан, указатель которого останавливается на призе в суперигре (суперпризе). Ведущий читает задание для суперигры. На табло слово (фраза) на тему, определенную в задании. Игроку дается право назвать несколько букв. Если названные буквы есть в задании, то ассистент ведущего выходит и открывает эти буквы на табло. Ведущий предлагает игроку ответить сразу или взять минуту на размышление. Если взята минута, то по ее истечении ведущий просит дать ответ. Игрок может отказаться от ответа (из-за отсутствия даже версий ответа). Если ответ верный, то ведущий поздравляет игрока с победой и вручает ему суперприз. Если игрок отказался отвечать или ответ неверный, то он теряет все свои призы (не только суперприз, но и те призы, которые он выиграл до суперигры), и игра заканчивается.

Диаграмма типа eEPC, построенная на основе этого описания, приведена на рис.8.3.4.

На рис.8.3.5 изображена eEPC-диаграмма, детализирующая функцию “Игра в туре” из диаграммы на рис.8.3.3. На рис.8.3.5 изображена eEPC-диаграмма, соответствующая следующему сценарию игры в каждом из трех туров и финале.

После объявления начала игры или очередного тура (финала) ведущий представляет тройку игроков, которые занимают свои места за барабаном. Затем ведущий читает задание, в котором предлагается тема и слово по этой теме.

Очередному игроку тройки ведущий предлагает назвать слово целиком или крутить барабан. Во втором случае, т. е. пока крутится барабан, ведущий беседует с игроком и, например, предлагает рассказать о себе, передать привет и т. д. Игрок (при желании) может выступить, вручить подарок ведущему и передаче.

Если указатель барабана остановился на секторе с очками, то ведущий называет количество выпавших очков на барабане и предлагает игроку назвать букву или слово. Игрок называет букву или слово целиком. Когда названной буквы в слове нет, тогда ведущий объявляет переход хода к следующему игроку, и с ним повторяется тот же цикл. Когда игрок называет букву и в слове она есть, тогда ассистент по просьбе ведущего выходит в студию и открывает эту букву или несколько таких букв в слове на табло. Если после этого остались еще не открытые буквы, то ведущий предлагает игроку снова крутить

барабан или назвать слово целиком. В случае угадывания игроком подряд трех правильных букв ведущий предлагает ассистенту внести в студию две шкатулки, и начинается игра “Две шкатулки”. Если игрок называет правильное слово, то ведущий объявляет победителя очередного тура и может через своих ассистентов вручить всем игрокам тура подарки. Когда слово названо неверно, игрок по просьбе ведущего занимает место среди зрителей, и игра продолжается с оставшимися игроками. Игра в туре заканчивается, если названо верное слово или если за барабаном не осталось ни одного игрока.

В предыдущем абзаце мы в целом описали ход игры, если на барабане выпал сектор с очками или игрок отгадал подряд три буквы. Детали игры в этих и других возможных случаях раскрываются в eEPC-модели функции “Обработка выпавшего сектора” (смотрите рис.8.3.5).

В финале (четвертом туре) участвуют победители трех отборочных туров. Игра проходит по правилам отборочных туров. После объявления победителя подсчитываются его набранные очки на барабане и предлагается на эту сумму перечень призов. Игрок выбирает призы. Ведущий предлагает победителю суперигру. Если игрок отказывается, то он забирает призы, и ведущий объявляет игру законченной.

В диаграмме на рис.8.3.5 функция “Обработка выпавшего сектора” является сложной. Ее детализация содержит модели таких подпроцессов как “Две шкатулки”, “Приз в студию” и т. д.

Рекомендуем читателям самостоятельно построить eEPC-диаграммы на основе своих наблюдений игры “Поле чудес” или следующих описаний сценариев ее составных частей (подпроцессов).

После окончания одного из туров (обычно после второго) может начаться игра со зрителями, в которой ведущий, объявляя тему, предлагает зрителям отгадать слово на табло. Зрители выявляют желание назвать слово поднятием руки. Если есть желающие назвать слово, то ведущий выбирает зрителя и предлагает ему назвать слово. Когда зритель называет правильное слово, ведущий объявляет его победителем и приглашает на площадку для вручения приза. Ведущий просит зрителя представиться и вручает приз. В случае неверного ответа ведущий передает слово другому поднявшему руку зрителю. Когда несколько человек подряд называли слово неверно или не осталось желающих назвать слово, ведущий открывает одну букву в слове на табло, и игра продолжается. Если слово никто не отгадал, и ведущий не считает нужным открыть очередную букву, то он объявляет зрителей проигравшими и открывает слово на табло.

При условии, что у игрока выпал сектор “Приз”, игроку

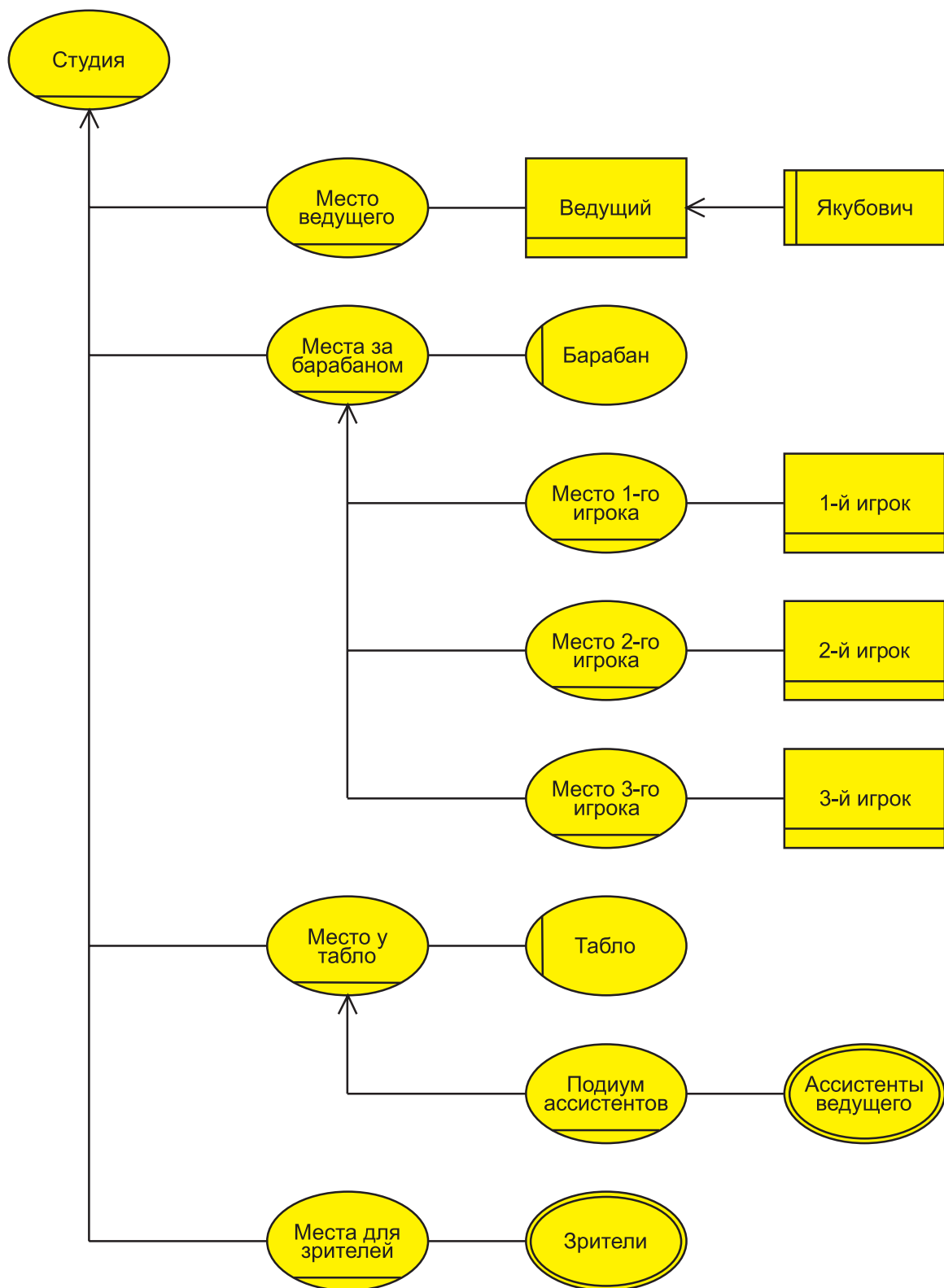


Рис.8.3.1 Диаграмма рабочих мест в студии игры "Поле Чудес"

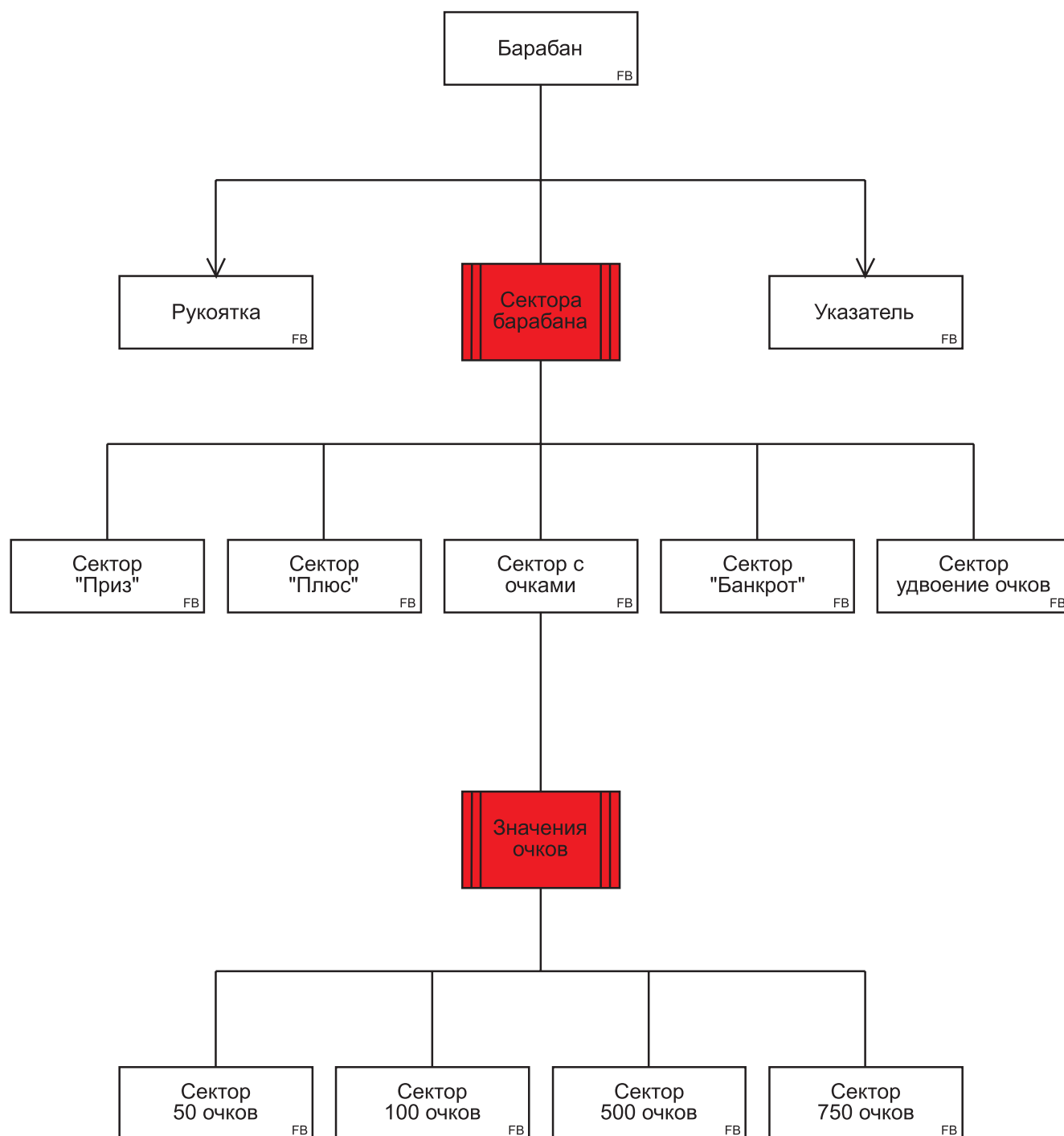


Рис.8.3.2 Диаграмма понятий, связанных с термином "Барабан"

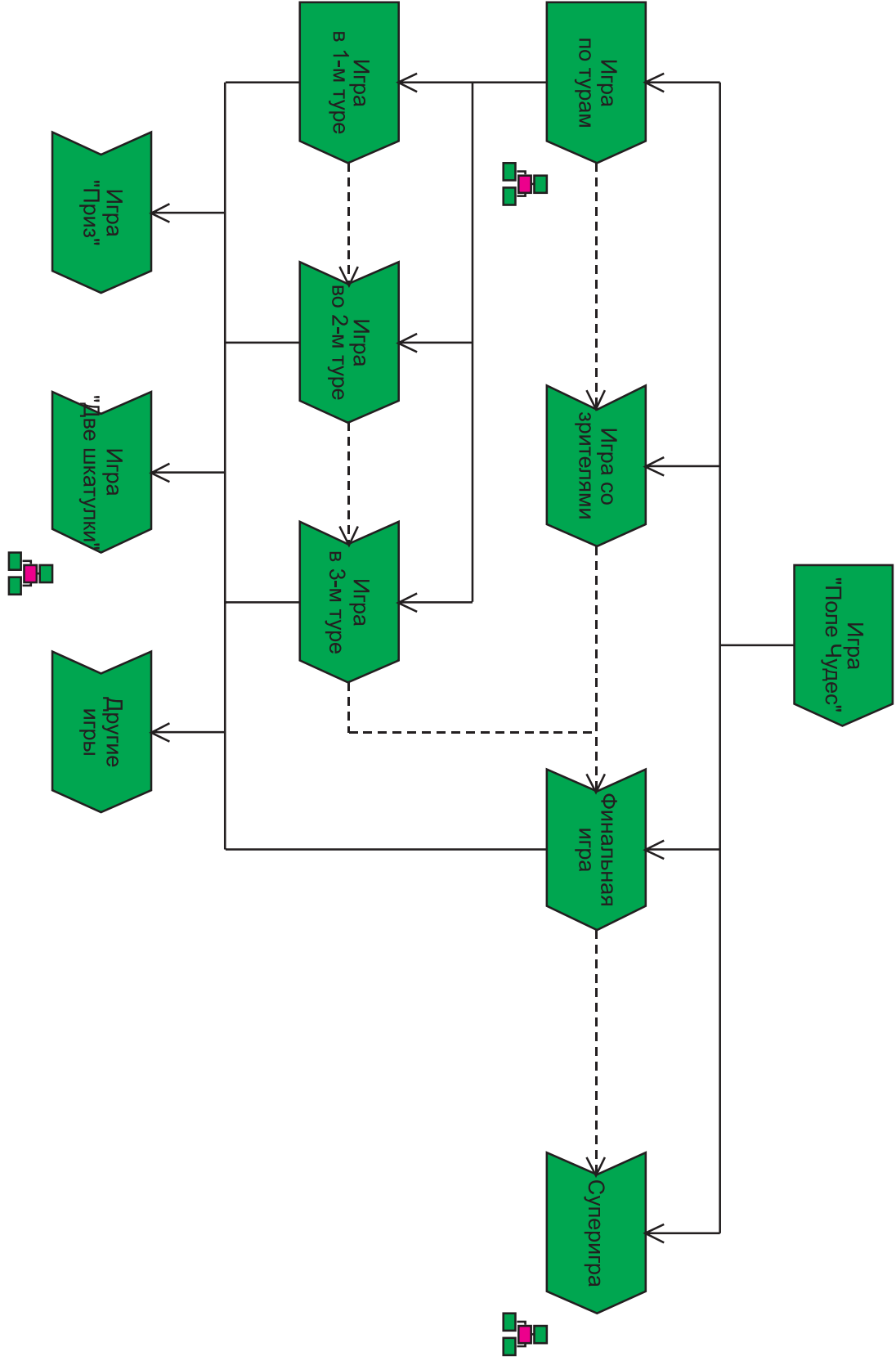


Рис.8.3.3 Диаграмма основных процессов игры "Поле Чудес"

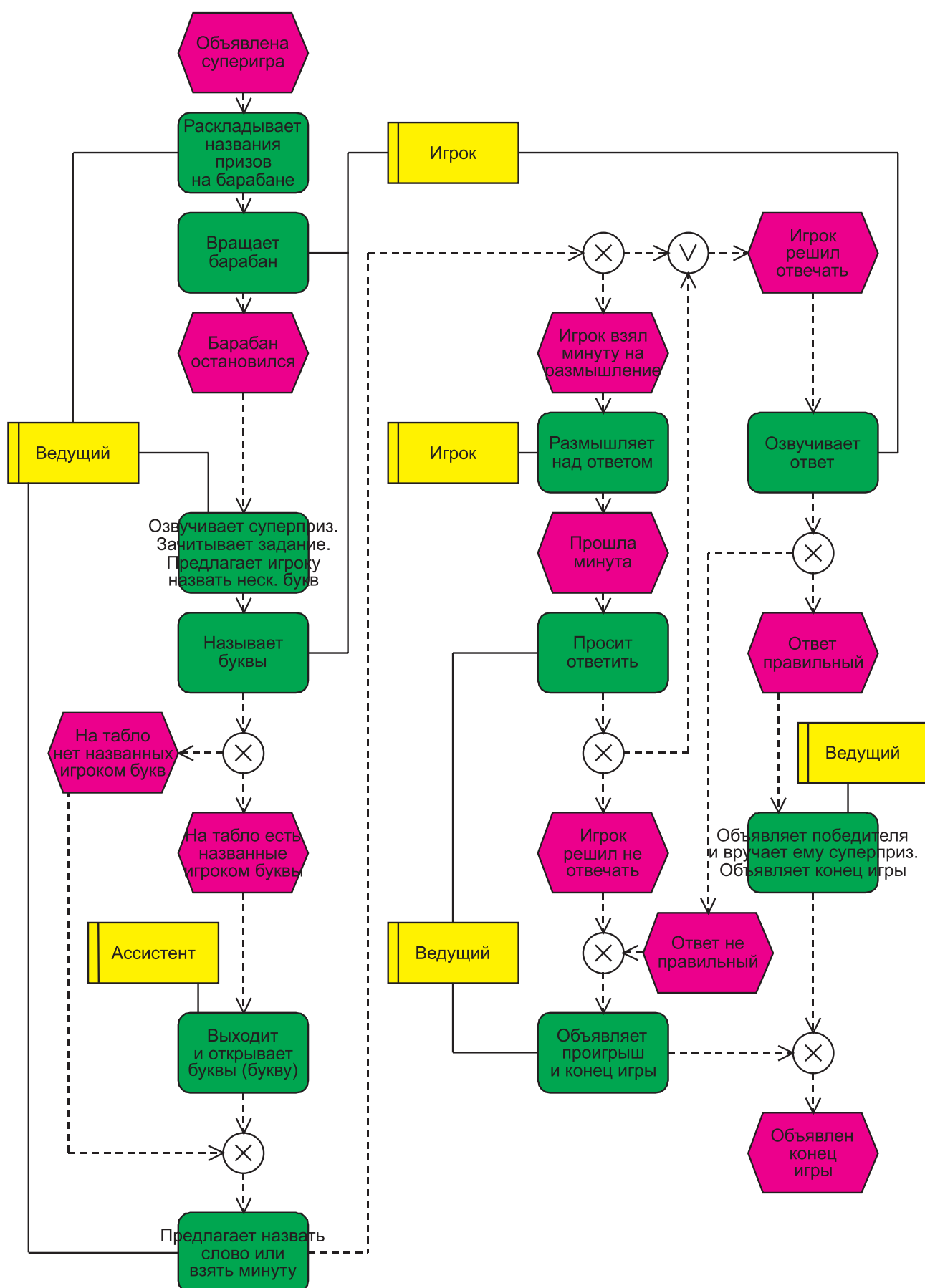


Рис.8.3.4 eEPC-диаграмма суперигры



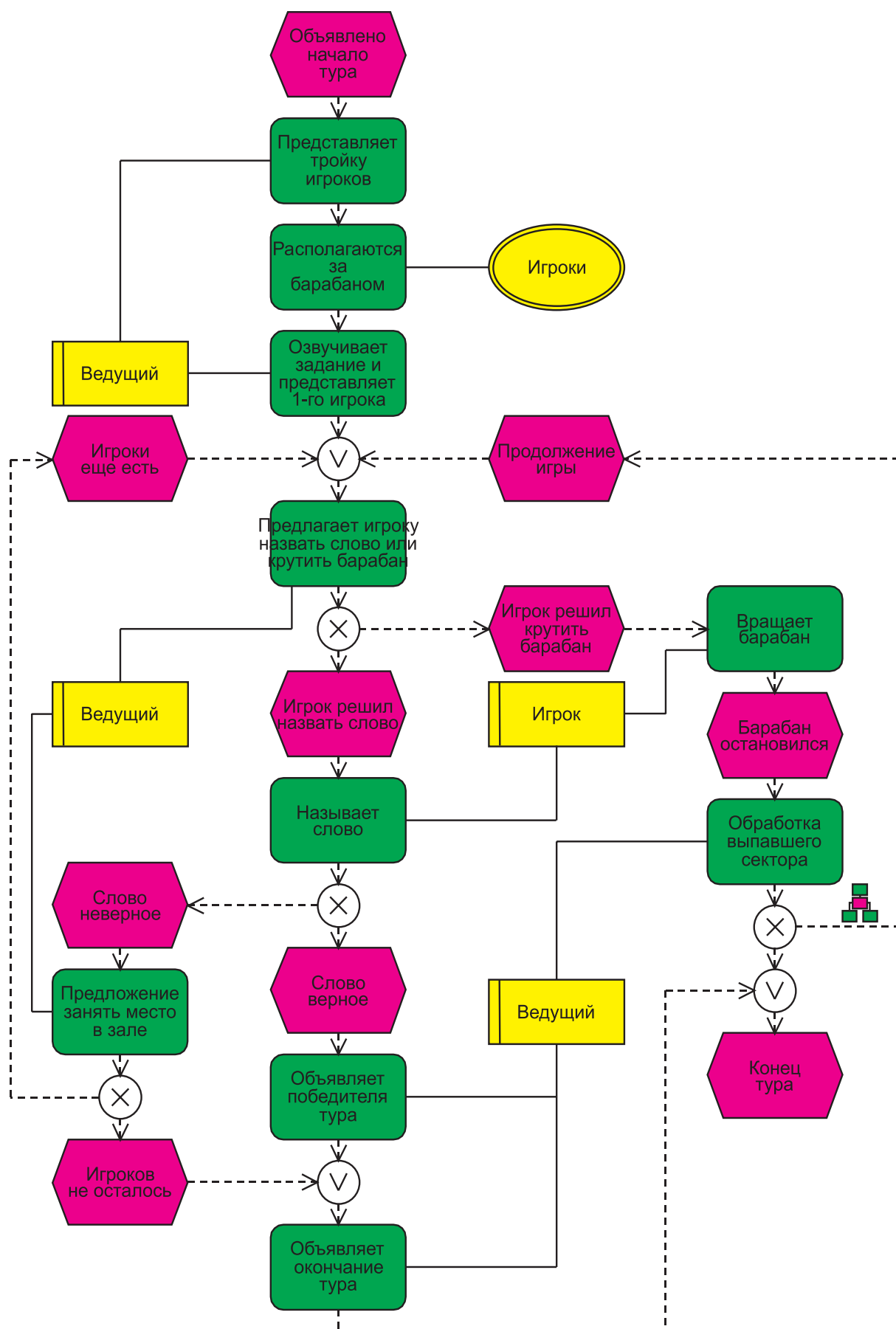


Рис.8.3.5 Диаграмма процесса игры в туре

предлагается выбрать приз и выйти из дальнейшей игры или отказаться от приза и продолжить игру. Если игрок отказался от приза, то он имеет право назвать букву или слово, т. к. сектор “Приз” – это результативный ход. Если игрок выбрал приз, то ведущий приглашает ассистента вынести ящик. Вокруг содержимого этого ящика начинается торг. Игроку предлагается взять определенную сумму денег или приз в ящике. В случае отказа от денег ведущий увеличивает сумму (до некоторого известного ему предела) или сразу открывает ящик для вручения приза. Если ведущий увеличивает сумму, а игрок все равно отказывается от денег, то в конце концов ведущий открывает ящик. Игроку вручается приз, и он покидает свое место за барабаном. Если игрок соглашается взять деньги, то ведущий предлагает ассистенту вынести деньги, которые вручаются игроку, и он покидает свое место за барабаном. По желанию игрока ведущий после этого может показать ему и зрителям приз в ящике.

Игра “Две шкатулки” – это один из подпроцессов, детализирующих функцию “Обработка выпавшего сектора”. Она проходит по следующему сценарию.

Если игрок назвал подряд правильно три буквы в угадываемом слове, то ведущий приглашает ассистента вынести в студию поднос с двумя закрытыми шкатулками. Игроку предлагается угадать, в какой из них деньги. Игрок делает выбор. Если в выбранной шкатулке деньги есть, то ведущий показывает их игроку и зрительному залу, поздравляет игрока и вручает ему эту сумму денег. Если выбранная шкатулка пуста, то ведущий показывает ее игроку и зрительному залу, а также открывает и показывает другую шкатулку, где действительно лежали деньги. Игрок в этом случае ничего не получает. При обоих исходах игрок остается за игровым столом и продолжает игру. На рис.8.3.6 изображена eEPC-диаграмма этого процесса.

В заключение коротко опишем (точнее, напомним) игру “Что? Где? Когда?”. В игре “Что? Где? Когда?” участвуют шесть знатоков, крупье и одиннадцать телезрителей, которые задают вопросы знатокам. Ассистент крупье раскладывает письма телезрителей на игровом столе в отдельные сектора. Игра начинается по звуку гонга. После вращения указатель останавливается на одном из секторов. Если на секторе указана сумма денег, то крупье объявляет телезрителя, который задает вопрос на телеэкране или в игровом зале. От имени и по поручению этого телезрителя вопрос может задать сам крупье. Если выпадает сектор ZERO, то крупье обсуждает с игроками сумму выигрыша и сложность вопроса. После сформулированного вопроса дается минута на размышление. Знатоки отвечают сразу (зарабатывая дополнительную минуту для использования в следующих турах) или думают и активно обсуждают

свои идеи ответа одну минуту. По истечении одной минуты крупье просит назвать игрока, который сформулирует ответ на поставленный вопрос. Капитан команды знатоков отвечает сам или объявляет знатока, который будет отвечать на вопрос. При отсутствии даже версий ответа (редкое событие для российских знатоков) игроки могут отказаться от ответа. Если ответ на поставленный вопрос верен, то крупье поздравляет команду с удачным туром, выигрыш в котором переходит на счет команды игроков. Если команда отказывается отвечать или ответ неверный, то они проигрывают тур, одно очко и соответствующие деньги. Если игроки первыми набирают шесть очков, то игра выиграна знатоками, в противном случае – телезрителями. Игра заканчивается.

Рекомендуем читателям формально описать организацию и проведение игры “Что? Где? Когда?”.

#### **8.4. Разработка технической документации**

На многих предприятиях военно-промышленного комплекса России происходит конверсия, т. е. переориентация на изготовление продукции мирного назначения. Одно из таких предприятий периодически получает заказы (в том числе от иностранных фирм) на выведение искусственных спутников на орбиту со своих ракетополетителей. Процесс обработки такого заказа можно условно разделить на два крупных подпроцесса – разработка и проектирование изделия, изготовление и проверка изделия.

Если поступило предложение на размещение вышеописанного заказа, то начинается процесс разработки и проектирования изделия, который включает в себя разработку конструкции изделия и расчет затрат на его изготовление. По окончании проектирования заключается договор. Это целый процесс, который включает в себя утверждение конструкции изделия (ее согласование с заказчиком), а также согласование условий финансирования между фирмой-заказчиком и предприятиями-изготовителями. Процесс изготовления и проверки изделия включает в себя разработку технической документации, изготовление материальной части и испытания.

Каждая из перечисленных функций является комплексной, т. е. состоит из множества других функций, подфункций и их составляющих. Поэтому остановимся подробнее только на процессе разработки технической документации.

На рис.8.4.1 изображена модель технических терминов, связанных с понятием “Техническая документация”. Далее приведены определения всех терминов этой модели.

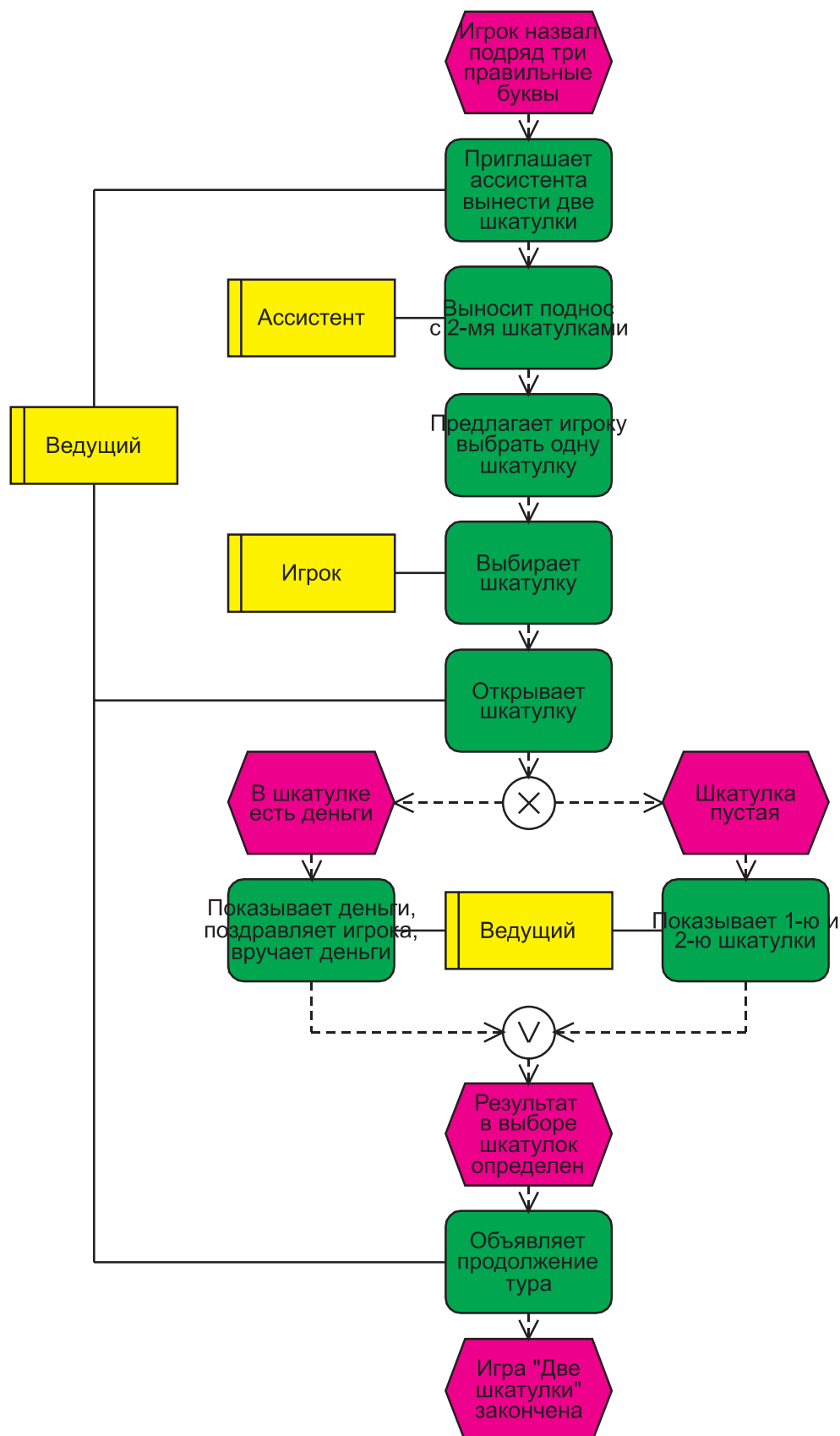


Рис.8.3.6 eEPC-диаграмма игры "Две шкатулки"

*Техническая документация* – документы, содержащие данные о составе и устройстве изделия, об особенностях его разработки и изготовления, а также описания применяемых при этом технологических процессов.

*Конструкторская документация* – комплект графических и текстовых документов, которые описывают состав и устройство изделия и содержат необходимые данные для его разработки, изготовления, контроля, приемки, эксплуатации и ремонта.

*Технологическая документация* – комплект документов, содержащих сводную информацию, необходимую для решения одной или комплекса инженерно-технических, планово-экономических или организационных задач. Это документы, применяемые при разработке, внедрении и функционировании технологических процессов и операций.

*Чертеж* – документ, содержащий изображение детали и другие данные, необходимые для ее изготовления и контроля.

*Общий проектный чертеж* – документ, определяющий конструкцию изделия, взаимодействие его составных частей и поясняющий принцип работы изделия.

*Рабочий чертеж* – документ, содержащий изображение сборочной единицы и другие данные, необходимые для ее сборки (изготовления) и контроля.

*Схема* – документ, на котором показаны в виде условных изображений или обозначений составные части изделия и связи между ними.

*Электрическая схема* – документ, на котором показаны в виде условных изображений или обозначений электрические элементы изделия и связи между ними.

*Пневматическая схема* – документ, на котором показаны в виде условных изображений или обозначений пневматические устройства и пневматические связи между ними.

*Инструкция* – документ, содержащий указания и правила, используемые при изготовлении (сборке, регулировке, контроле, приемке и т. д.) изделия.

*Инструкция по настройке* – документ, содержащий указания и правила, используемые при настройке изделия.

*Инструкция по эксплуатации* – документ, содержащий указания и правила по эксплуатации изделия.

*Инструкция по электропроверкам* – документ, содержащий указания и правила, используемые при проведении электропроверок.

*Документация технологических процессов* – документы, полностью и однозначно определяющие технологический процесс (операцию) изготовления или ремонта изделия (составных частей изделия).

*Технологические указания* – документы, предназна-

ченные для описания технологических процессов, методов и приемов, повторяющихся при изготовлении или ремонте изделий (составных частей изделий), правил эксплуатации средств технологического оснащения.

*График-маршрут* – документ, предназначенный для указания полного состава технологических операций при операционном описании изготовления или ремонта изделия (составных частей изделия).

На рис.8.4.2 изображена диаграмма подчиненности должностей (субъектов ответственности) конструкторского отдела фирмы. Из этой диаграммы можем увидеть, что начальнику конструкторского отдела подчиняются начальники двух секторов (конструкторского сектора и сектора схем). В непосредственном подчинении у начальника сектора находятся начальники групп. Начальнику конструкторского сектора подчиняются начальник конструкторской группы и начальник группы кабелей, начальнику сектора схем – начальник группы САПР (системы автоматического проектирования) и начальник группы схемотехников. В подчинении у начальника группы находятся специалисты различной квалификации. Это ведущий специалист, специалист первой категории, специалист второй категории, специалист третьей категории и специалист-техник. В конструкторской группе и группе кабелей специалистами являются конструкторы, в группе САПР – инженеры-математики, в группе схемотехников – инженеры.

Диаграмма основных процессов рассматриваемой фирмы изображена на рис.8.4.3, а на рис.8.4.4 изображен функциональный поток процесса разработки конструкторской документации. Из этой диаграммы видно, что после поступления плана-графика конструкторских работ начинается разработка общих проектных чертежей. Ее выполняет конструктор. По ее окончании общие проектные чертежи проверяются начальником конструкторского отдела. Если высказаны замечания, то конструктор дорабатывает общие проектные чертежи, и они вновь поступают на проверку начальнику конструкторского отдела. Если замечаний нет, то конструктор передает чертежи на изучение в смежные отделы. Затем идет их согласование с технологическими службами и смежными отделами. Если найдены расхождения в параметрах изделия, то конструктором вносятся коррективы, и чертежи вновь согласовываются. Если расхождений в параметрах изделия нет, то конструктор разрабатывает рабочие чертежи. После этого рабочие чертежи проверяются начальником конструкторского отдела. Если есть замечания, то конструктор дорабатывает рабочие чертежи, и начальник вновь их проверяет. Когда замечаний нет, технологом разрабатываются технологические про-

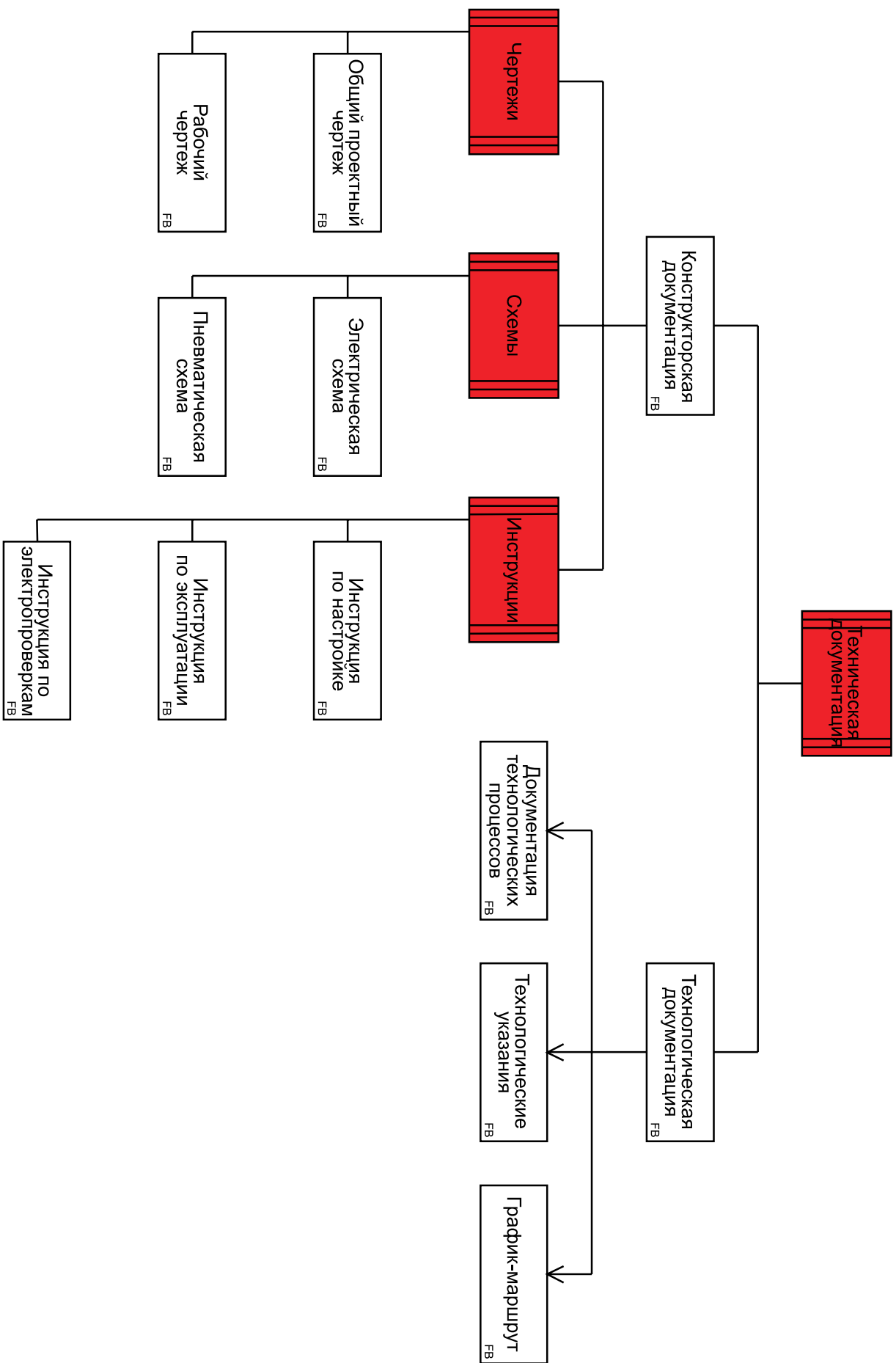


Рис. 8.4.1 Диаграмма терминов, относящихся к технической документации

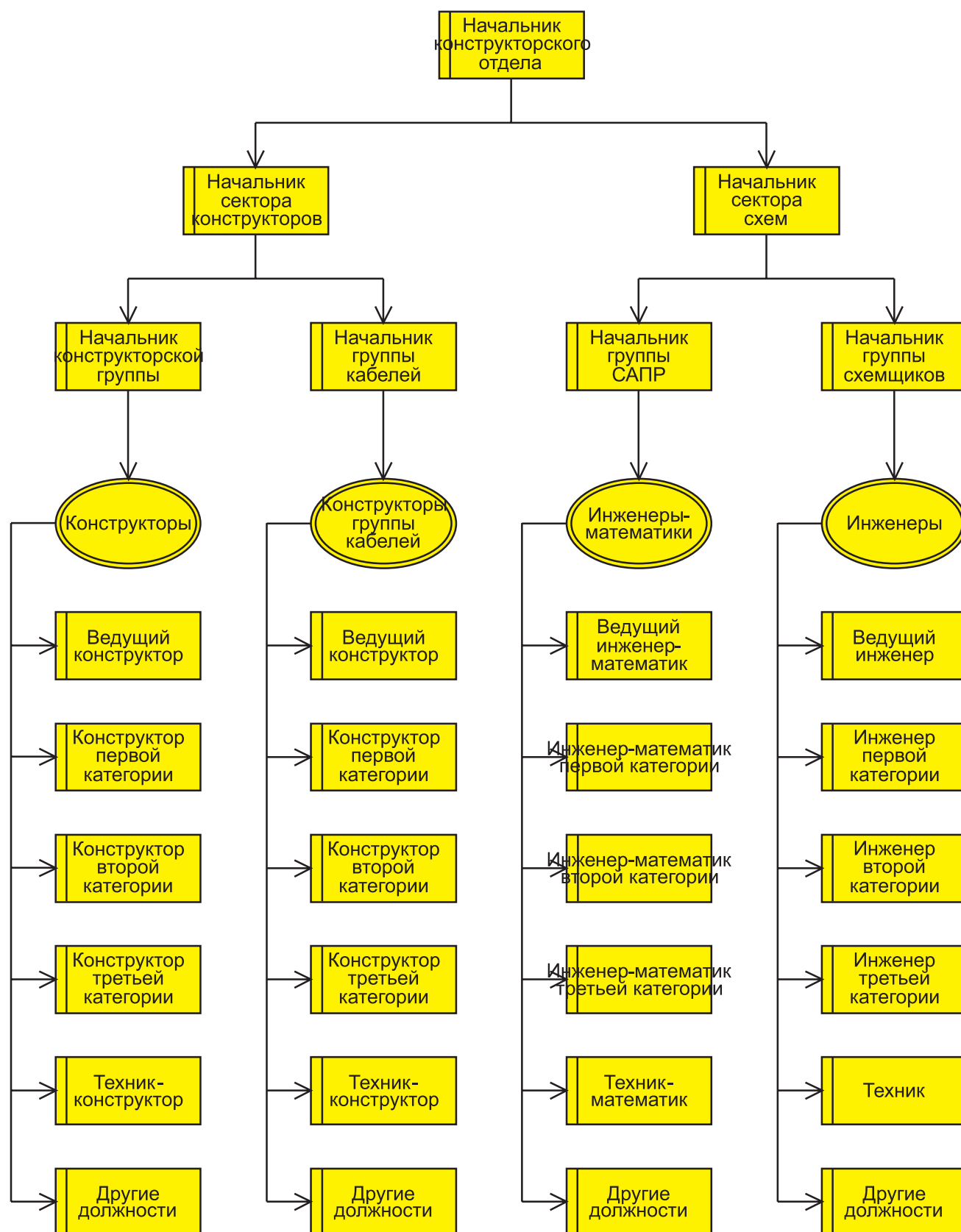


Рис.8.4.2 Иерархическое дерево должностей конструкторского отдела фирмы



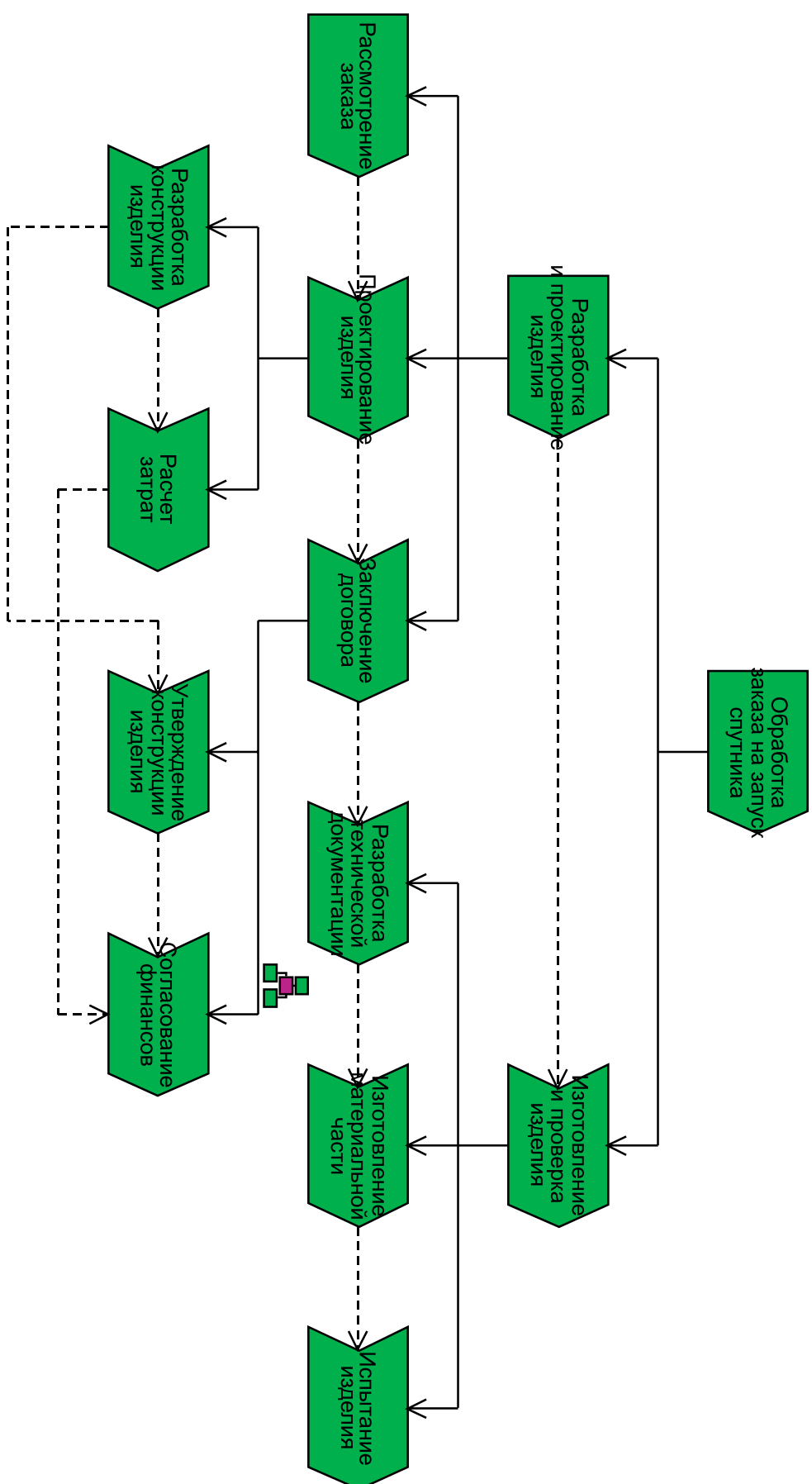


Рис.8.4.3 Диаграмма основных процессов фирмы



цессы. Затем рабочие изготавливают пробный образец, а технологи проводят его испытания. Если испытания не удались, то конструктор корректирует конструкторскую документацию. Затем технолог корректирует технологические процессы, рабочие дорабатывают пробный образец, и вновь проводятся испытания. Когда испытания удаются, начальник конструкторского отдела утверждает конструкторскую документацию.

### **8.5. Конкурсный отбор и виртуальная кафедра**

Рассмотрим факультет высшего учебного заведения, основные виды деятельности которого представлены в диаграмме на рис.8.5.1. Здесь над связями приведены названия их типов.

Подробное описание представленных видов деятельности факультета займет очень много места (более ста страниц). Поэтому ограничимся рассмотрением подпроцесса “Прием на работу профессорско-преподавательского состава (ППС)”. Этот подпроцесс является составляющим процесса “Администрирование ППС”, входящего в комплексный процесс “Управление факультетом”, который является первым в цепочке комплексных процессов, раскрывающих рабочие процессы на факультете (рис.8.5.1).

Процессно-ориентированное функциональное дерево процесса “Прием на работу ППС” изображено на рис.8.5.2. Это дерево отражает сложившиеся в ЮУрГУ и на электротехническом факультете (ЭТФ) процессы приема ППС на работу. В частности, если соискатель должности (СД) не имеет ученой степени, то он может претендовать только на должность ассистента или старшего преподавателя кафедры, и его конкурсное дело не рассматривается на совете ЮУрГУ.

Рассмотрим случай, когда СД избирается на должность на совете ЭТФ. Из рис.8.5.2 видно, что процесс “Избрание на должность на совете ЭТФ” состоит из шести функций. Последовательность этих функций в цепи процесса, их исполнители, а также запускающие и завершающие события для каждой функции изображены на рис.8.5.3. Все эти функции не являются элементарными и поэтому детализированы. На рис.8.5.4 приведена eEPC-диаграмма, детализирующая процесс допуска СД к участию в отборочном конкурсе. Здесь событие “Заявление не принято” наступает, например, если документы, подтверждающие квалификацию, знания и опыт СД не отвечают необходимым квалификационным требованиям вакантной должности. На рис.8.5.13 детализирована функция “Конкурсный

отбор СД на совете ЭТФ”. Диаграммы, детализирующие несложные функции “Объявление конкурса в печати”, “Подготовка к рассмотрению конк. дел на засед. кафедры” и “Подготовка к выборам на совете ЭТФ” здесь не представлены. Более сложная функция “Рассмотрение конк. дел на засед. каф.” детализирована на рис.8.5.8. Она входит в модель операционно-ориентированного дерева “Рассмотрение вопроса повестки засед. каф.”, изображенного на рис.8.5.7.

Процесс заседания кафедры изображен на рис.8.5.5. В этом процессе участвуют заведующий кафедрой, учёный секретарь кафедры и члены кафедры. После начала заседания кафедры ее учёный секретарь отмечает присутствующих. На основе этого анализируется кворум. Если кворума нет, то заведующий кафедрой выясняет причины срыва заседания кафедры и объявляет его не состоявшимся. Составляется пояснительная записка декану ЭТФ о причинах срыва заседания кафедры. Если кворум есть, то члены кафедры обсуждают и утверждают повестку заседания кафедры, которая заносится в протокол заседания. Затем последовательно рассматриваются вопросы повестки заседания кафедры. При этом ведется протокол с отражением хода рассмотрения вопросов повестки. Если все вопросы повестки заседания кафедры рассмотрены и запротоколированы, то заведующий кафедрой проводит процедуру утверждения протокола. Затем на его основе учёный секретарь готовит необходимые выписки в совет ЭТФ, деканат, ректорат и т.д. Распечатываются копии этих выписок. Заведующий кафедрой утверждает выписки, одновременно проводит закрытие заседания кафедры.

Этот же процесс представлен на рис.8.5.6 в виде офисной (пиктографической) диаграммы.

В модели на рис.8.5.7 перечислены только некоторые кадровые вопросы, которые могут рассматриваться на заседании кафедры. Из них на рис.8.5.8 детализировано только рассмотрение одного вопроса. Из этого рисунка видно, что заседание ведет заведующий кафедрой. В основные обязанности ученого секретаря кафедры входит ведение протокола заседания. Заведующий кафедрой представляет СД членам кафедры, которые приступают к обсуждению его кандидатуры. Если имеется несколько СД, то процедура обсуждения СД повторяется до тех пор, пока все СД, прибывшие на заседание кафедры, не будут рассмотрены. Если все СД рассмотрены, то членам кафедры предлагается проголосовать вопрос “Кто поддерживает СД?” по каждому из рассмотренных СД. После окончания голосования заведующий кафедрой формулирует рекомендации кафедры о СД. Ученый секретарь кафедры протоколирует текст сформулированных рекомендаций.

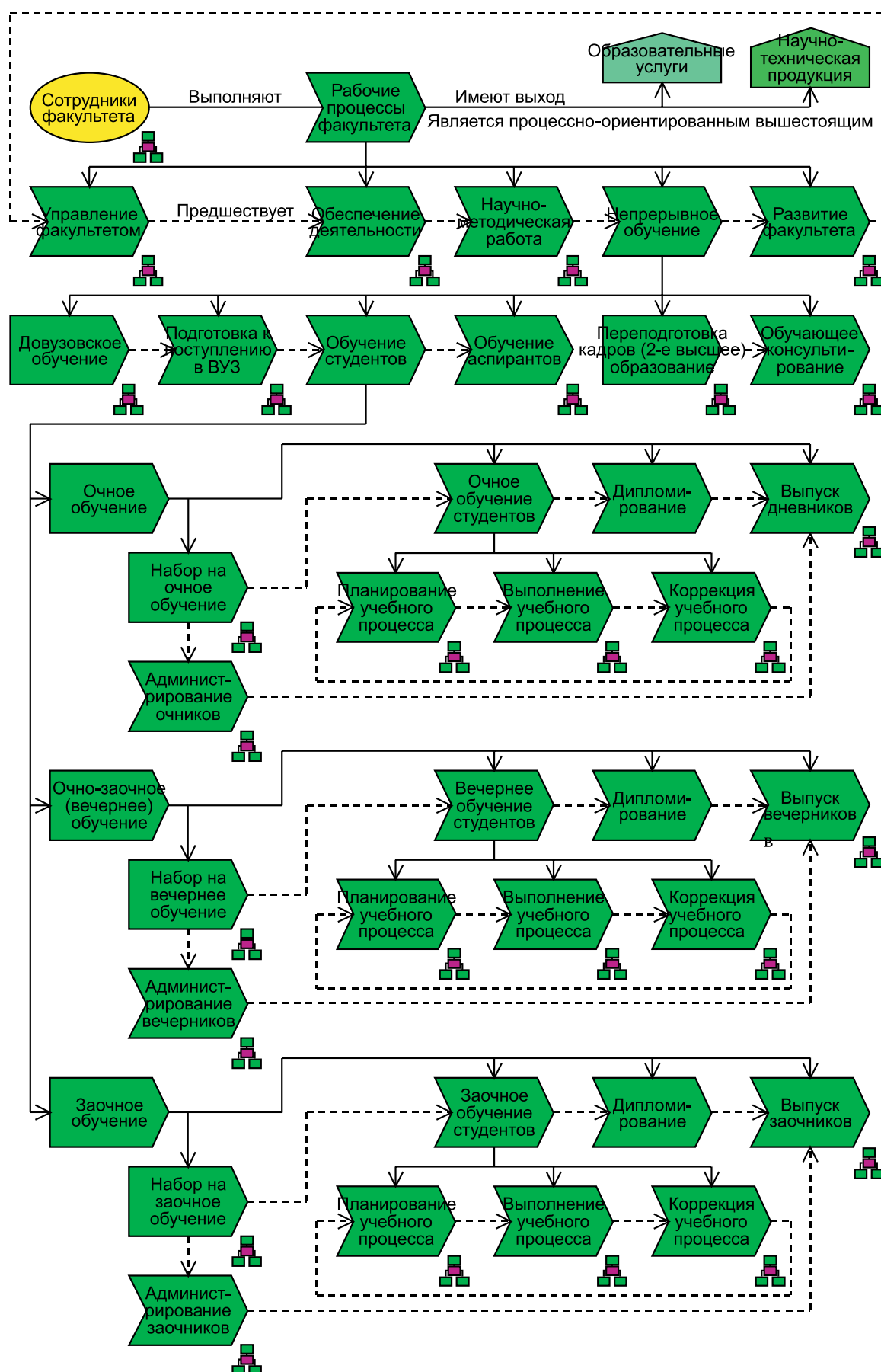


Рис.8.5.1 Основные процессы ЭТФ МФ ЮУрГУ

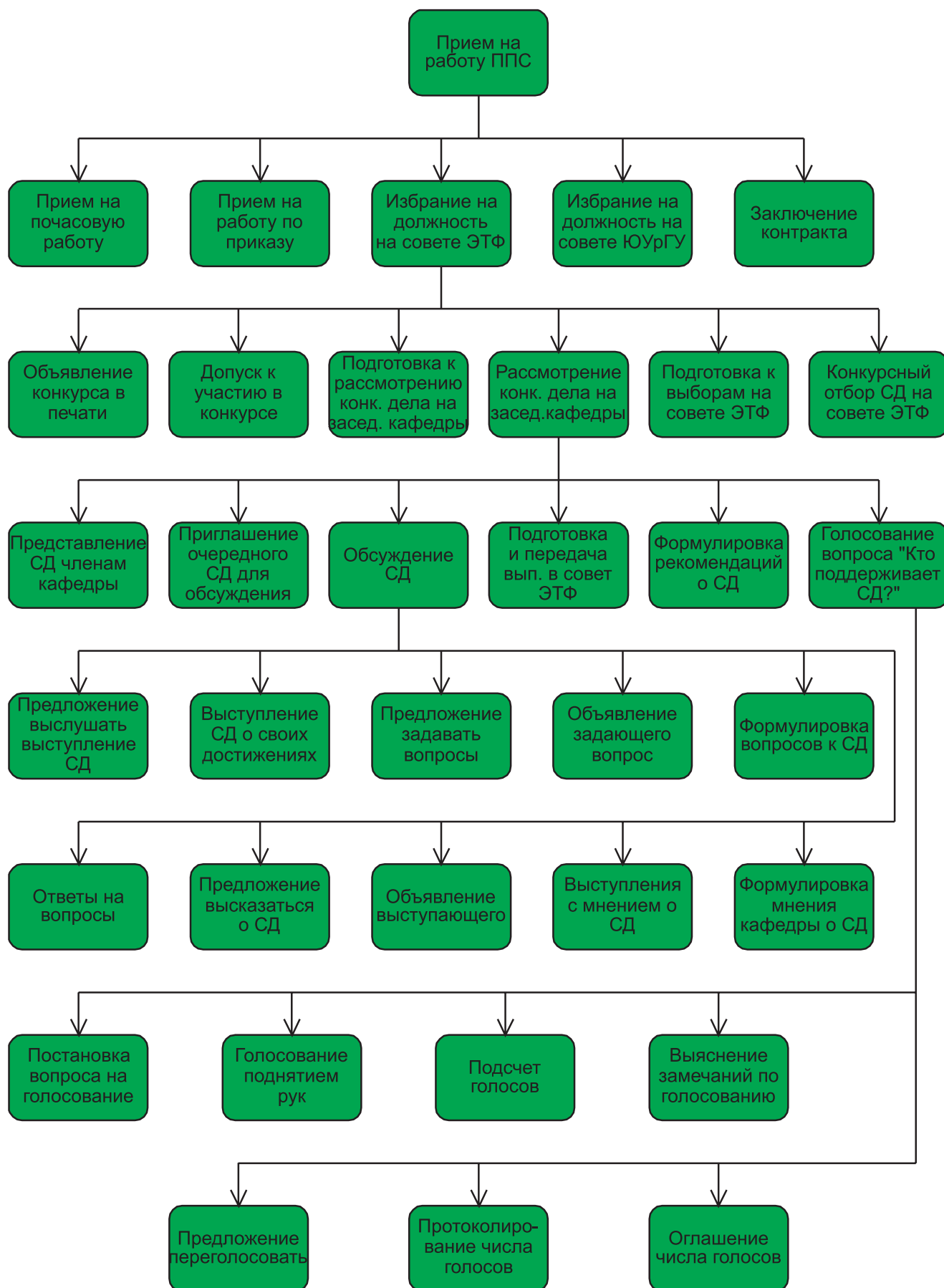


Рис.8.5.2 Функциональное дерево рабочего процесса “Прием на работу ППС”

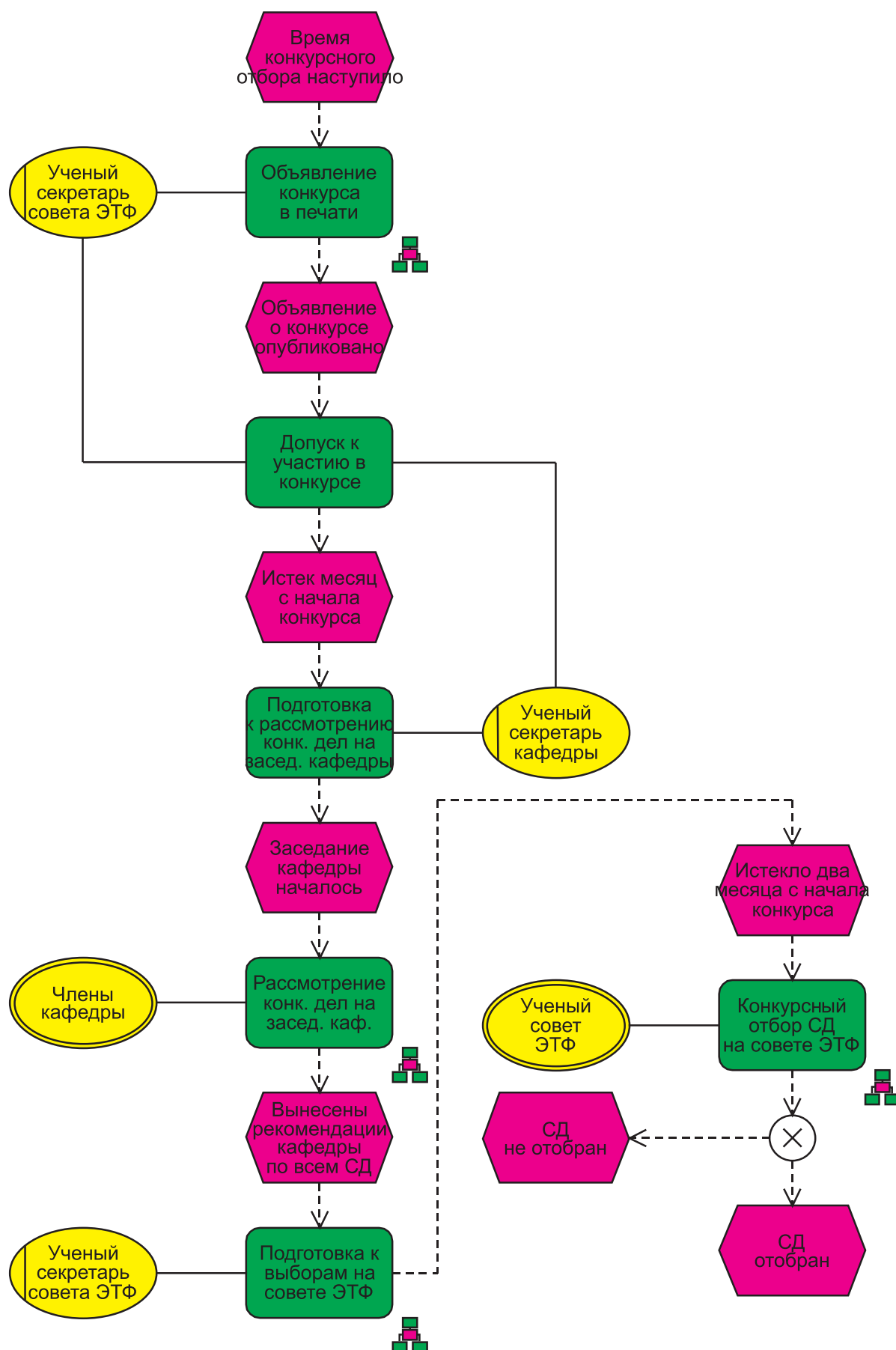


Рис.8.5.3 Функциональный поток процесса "Избрание на должность на совете ЭТФ"



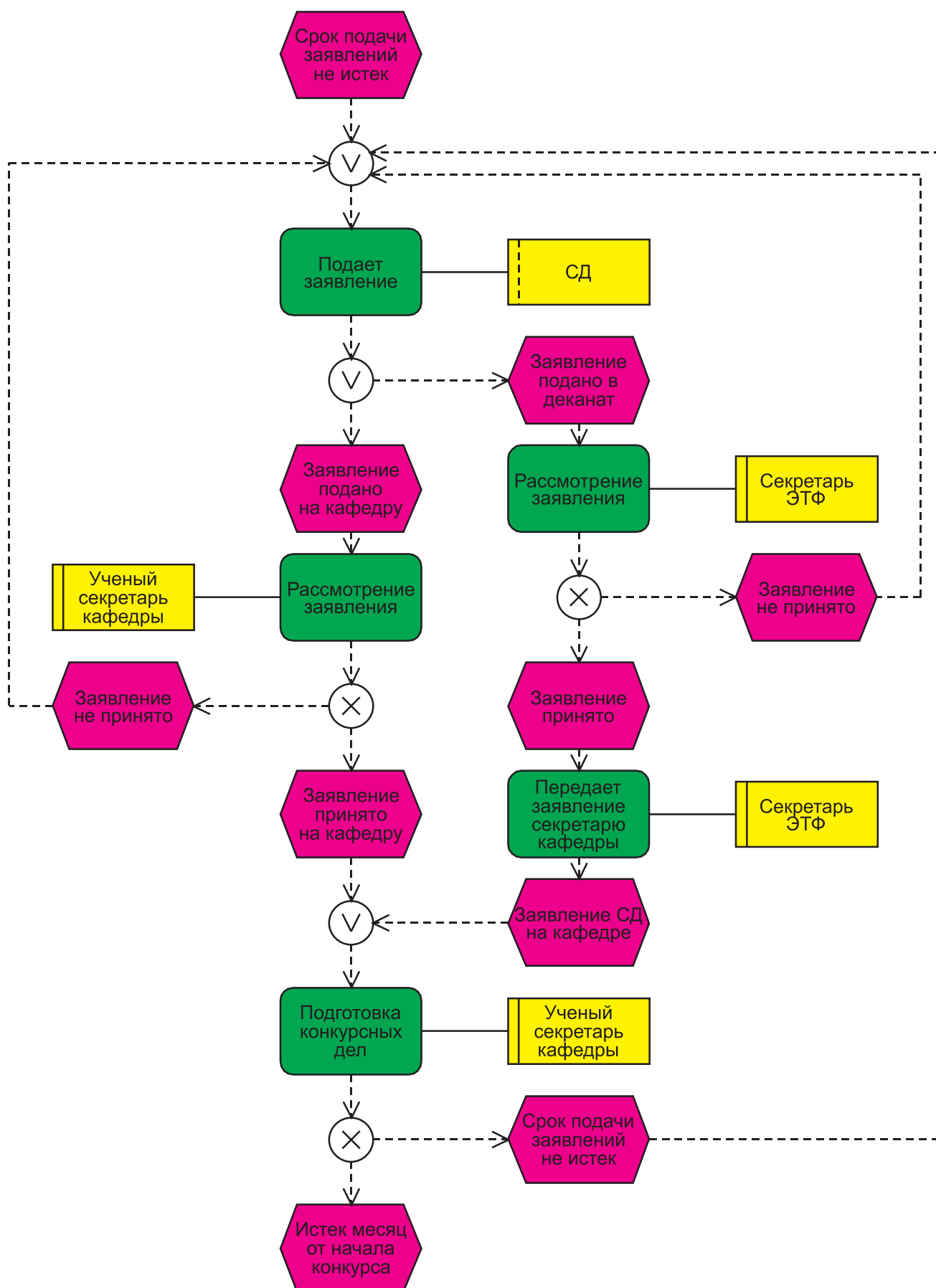


Рис.8.5.4 Функциональный поток процесса “Допуск к участию в конкурсе”

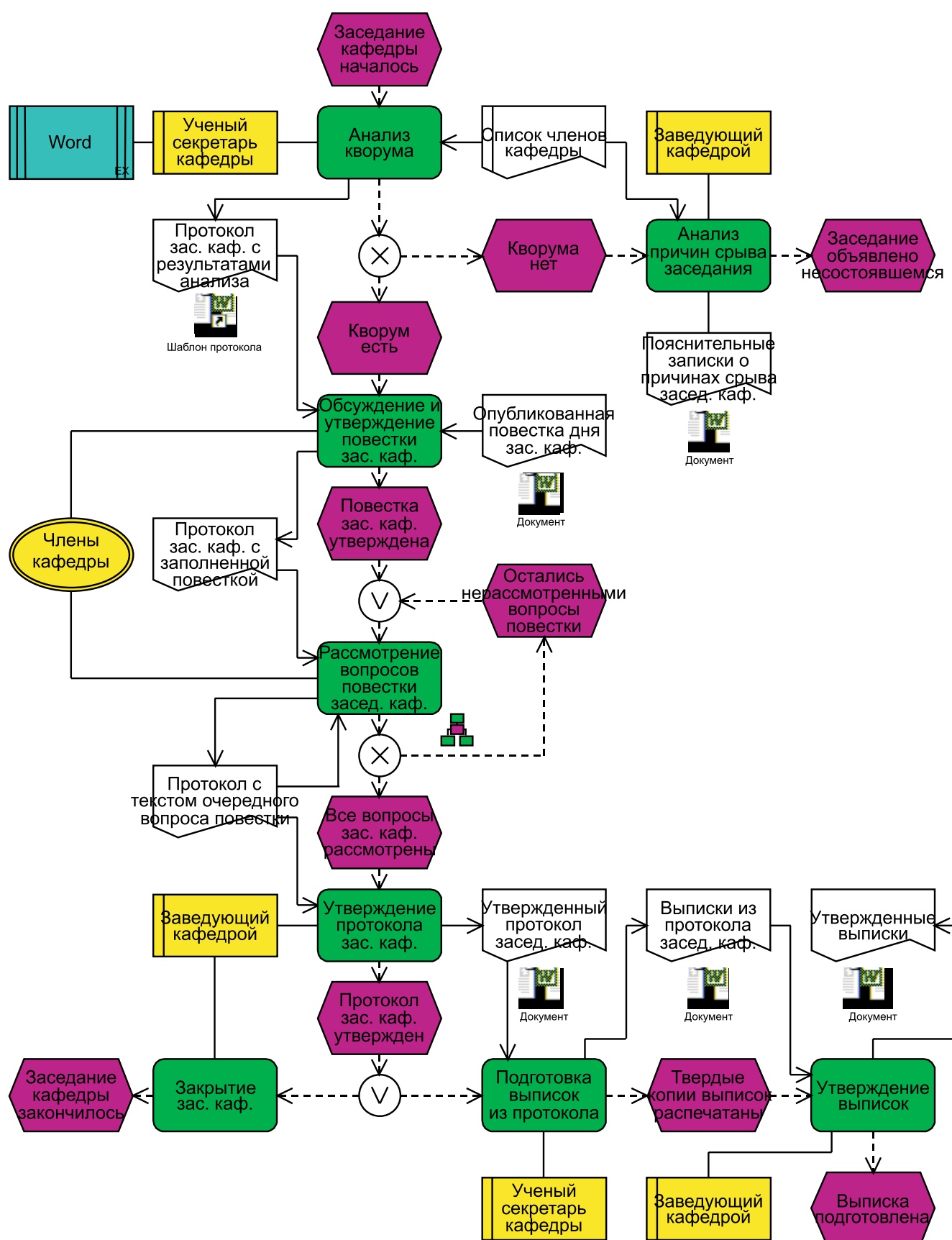


Рис.8.5.5 Функциональный поток процесса “Проведение заседания кафедры”

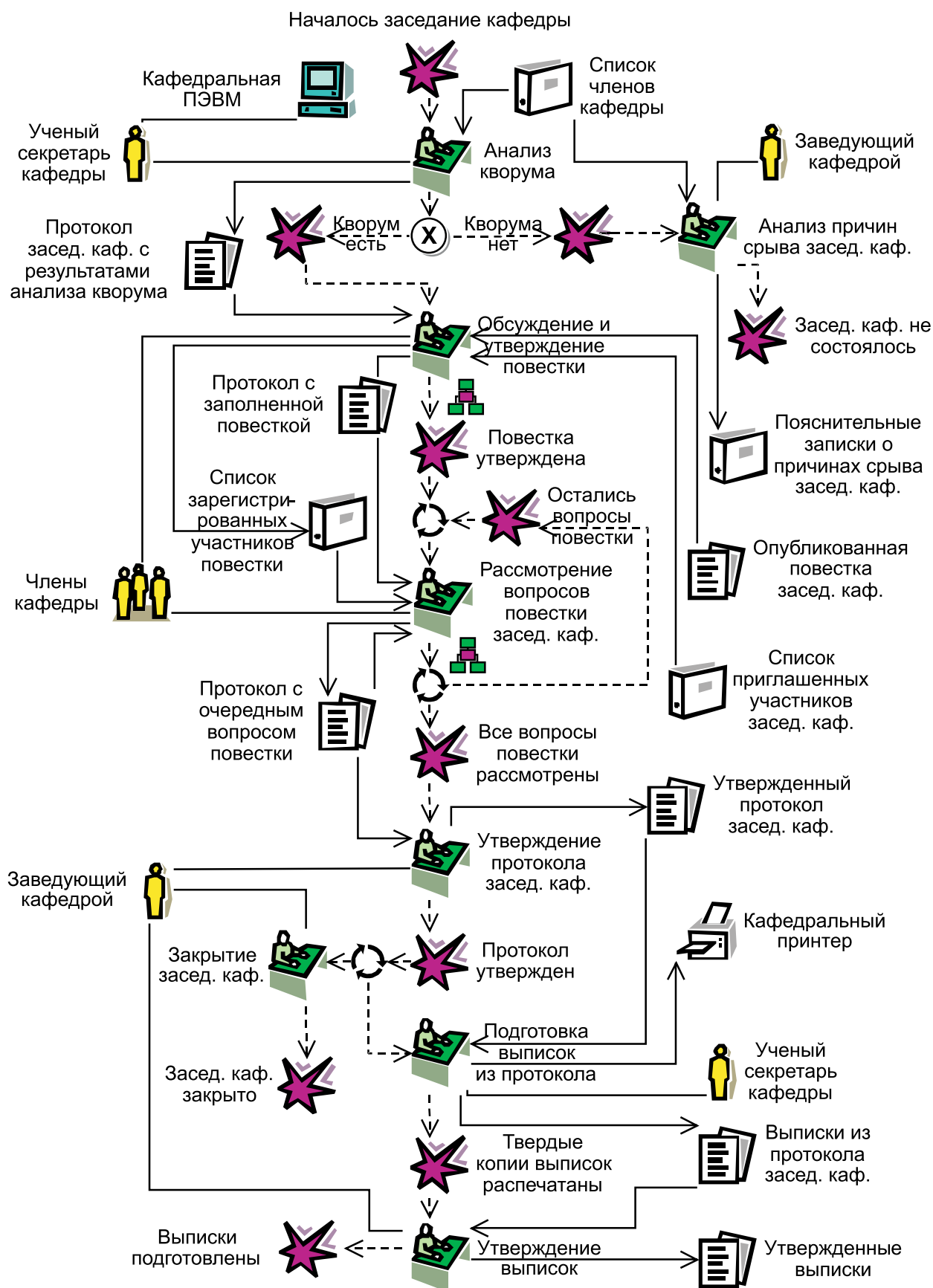


Рис.8.5.6 Функциональный поток офисного процесса “Проведение заседания кафедры”

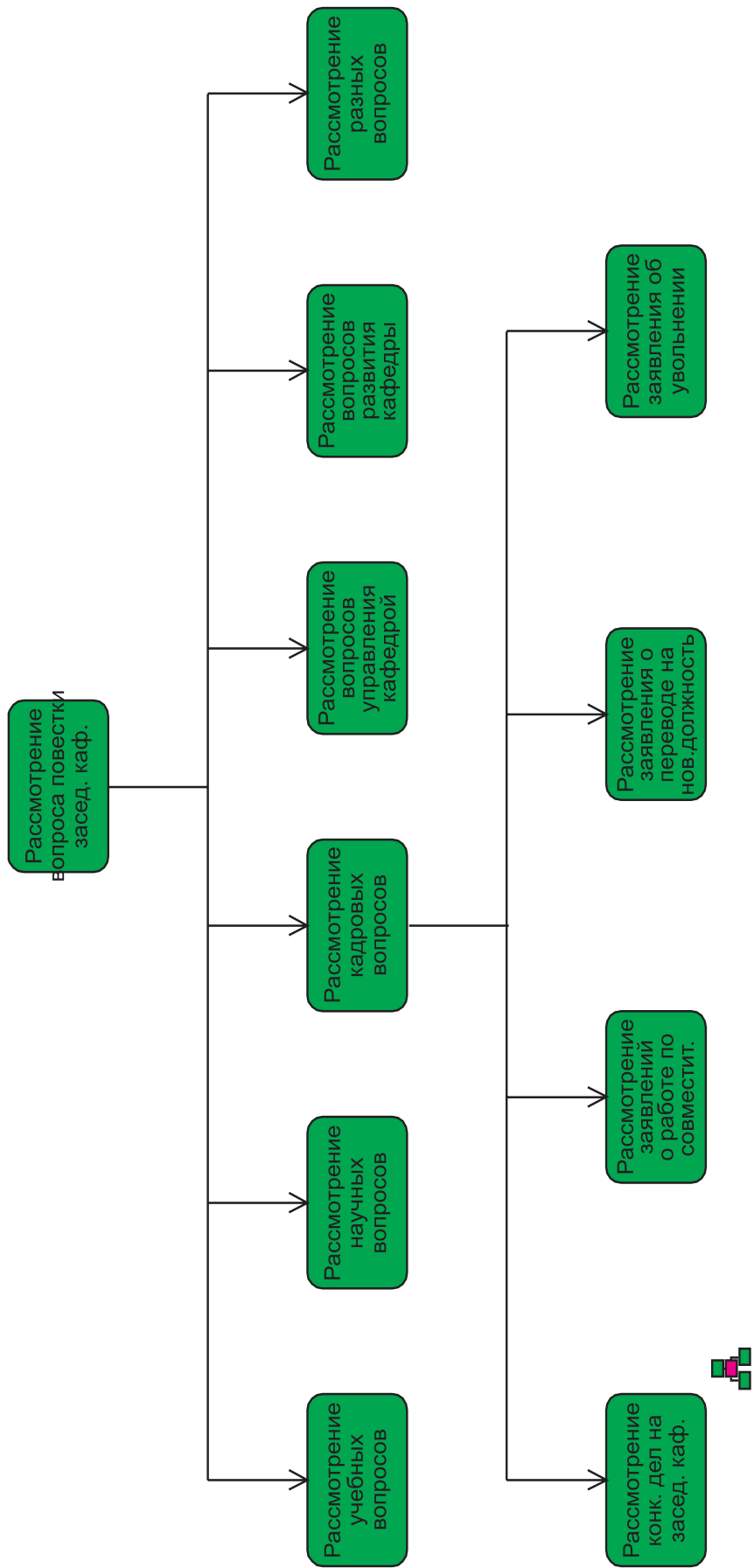


Рис.8.5.7 Операционно-ориентированное дерево “Рассмотрение вопроса повестки заседания кафедры”

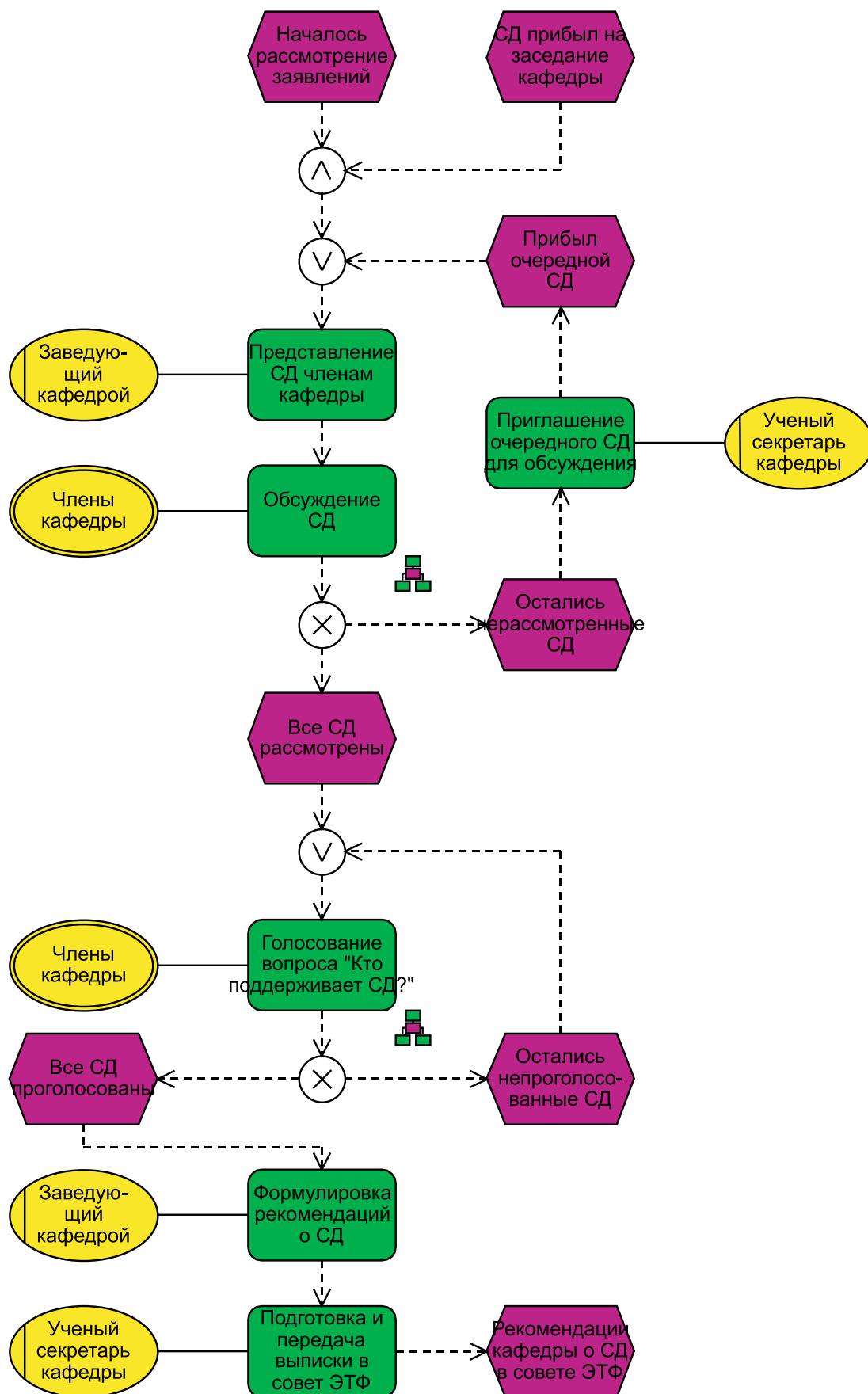


Рис.8.5.8 Функциональный поток процесса "Рассмотрение конкурсных дел на заседании кафедры"

Этот же процесс представлен на рис.8.5.9 в виде диаграммы офисного процесса. Очевидно, что эта диаграмма более наглядна и понятна.

На рис.8.5.10 детализирован процесс “Обсуждение СД” на заседании кафедры. В этом процессе по желанию членов кафедры СД должен выступать перед членами кафедры с докладом и отвечать на их вопросы. Как правило, доклад содержит сведения об основных достижениях СД. Выступая с докладом, СД стремится сформировать наилучшее мнение членов кафедры о своих деловых качествах. Заведующий кафедрой предлагает членам кафедры послушать выступление СД. Если члены кафедры знакомы с СД или информированы о его основных достижениях, то они могут отказаться от заслушивания доклада СД. Если члены кафедры желают прослушать доклад СД, то он выступает перед ними с докладом о своих достижениях, при этом основные моменты выступления СД заносятся в протокол ученым секретарем кафедры. Затем заведующий кафедрой предлагает задать вопросы СД. У членов кафедры может не возникнуть вопросов к СД. Если же вопросы возникли, то заведующий кафедрой объявляет члена кафедры, который формулирует вопрос к СД. При этом ученый секретарь кафедры продолжает вести протокол и заносит в него формулировку вопроса и ответ СД на этот вопрос. Если члены кафедры задали все свои вопросы или вопросов больше не возникло, то заведующий кафедрой предлагает приступить к высказыванию мнений о СД. Желающие члены кафедры поочередно высказывают свое мнение о СД. При этом заведующий кафедрой предварительно объявляет выступающего, а ученый секретарь заносит в протокол его фамилию и инициалы. В процессе выступления ученый секретарь протоколирует основные моменты (тезисы) выступления. После того, как все желающие высказали свое мнение о СД, заведующий кафедрой подводит итог этих выступлений, т. е. формулирует мнение кафедры о СД, а ученый секретарь заносит его в протокол. На этом процесс “Обсуждение СД” считается законченным. Результатом этого процесса является сформулированное мнение кафедры о СД и протокол как официальный документ, содержащий сведения об основных моментах данного процесса.

На рис.8.5.11 приведена eEPC-диаграмма процесса “Голосование вопроса”. Это типовой кафедральный процесс, который не зависит от формулировки голосуемого вопроса. Если в этой модели заменить исполнителей “Заведующий кафедрой”, “Члены кафедры” и “Ученый секретарь кафедры” соответственно на “Ведущий собрание”, “Члены собрания” и “Секретарь собрания”, то мы максимально обобщим этот процесс. На рис.

8.5.12 этот процесс представлен в виде диаграммы офисного процесса. Заметим, что здесь первый и последний логические операторы конкретизированы, а все остальные нет. Такое допускается во всех моделях с использованием пиктограмм.

В офисном процессе на рис.8.5.12 ведущий собрание ставит вопрос на голосование. Участники собрания голосуют поднятием рук. Ведущий собрания подсчитывает голоса. Затем секретарь собрания выясняет замечания по голосованию. Если замечания имеются, то секретарь собрания предлагает участникам собрания переголосовать. В противном случае секретарь собрания протоколирует число голосов, а затем оглашает протокольную запись участникам собрания.

Кратко остановимся на идеях создания виртуальных и распределенных по всей территории России кафедр, факультетов и в целом учебных заведений. Эти идеи обсуждаются на конференциях, в журнальных статьях и уже делаются попытки реализации подобных идей и концепций.

В первую очередь следует отметить, что дистанционное обучение широко внедряется в практику высших учебных заведений. Теперь профессор (преподаватель), читая лекцию, может находиться не в аудитории, а в своем офисе или даже в своем домашнем рабочем кабинете. Если к этому добавить перевод всех кафедральных процессов в e-процессы, то можно говорить о создании виртуальной кафедры. Профессора, преподаватели и сотрудники такой кафедры могут проживать в любом российском городе или за рубежом и быть полноправным и активным ее членом, т. е. избираться на должность, обсуждать учебные планы, участвовать в распределении учебной нагрузки, посещать занятия своих коллег по кафедре, если последние проводятся дистанционно.

Пример перевода одного из рассмотренных кафедральных процессов в e-процесс рассмотрен в конце этого раздела. Но прежде отметим, что переход от классических офисных БП к e-процессам позволяет снизить различного рода затраты, повысить эффективность использования всех ресурсов и улучшить деятельность организации в целом. Это происходит за счет снижения количества участников БП, автоматизации многих функций БП, повышения производительности, сокращения времени на выполнение каждой функции БП и т. д. Но самое главное – участники e-процесса могут находиться у себя дома, заниматься различными делами, в том числе участвовать в других e-процессах. Необходимо только одно – рядом должен быть компьютер с выходом в Internet и соответствующее программное обеспечение (ПО).

Сокращение численности участников БП позволяет



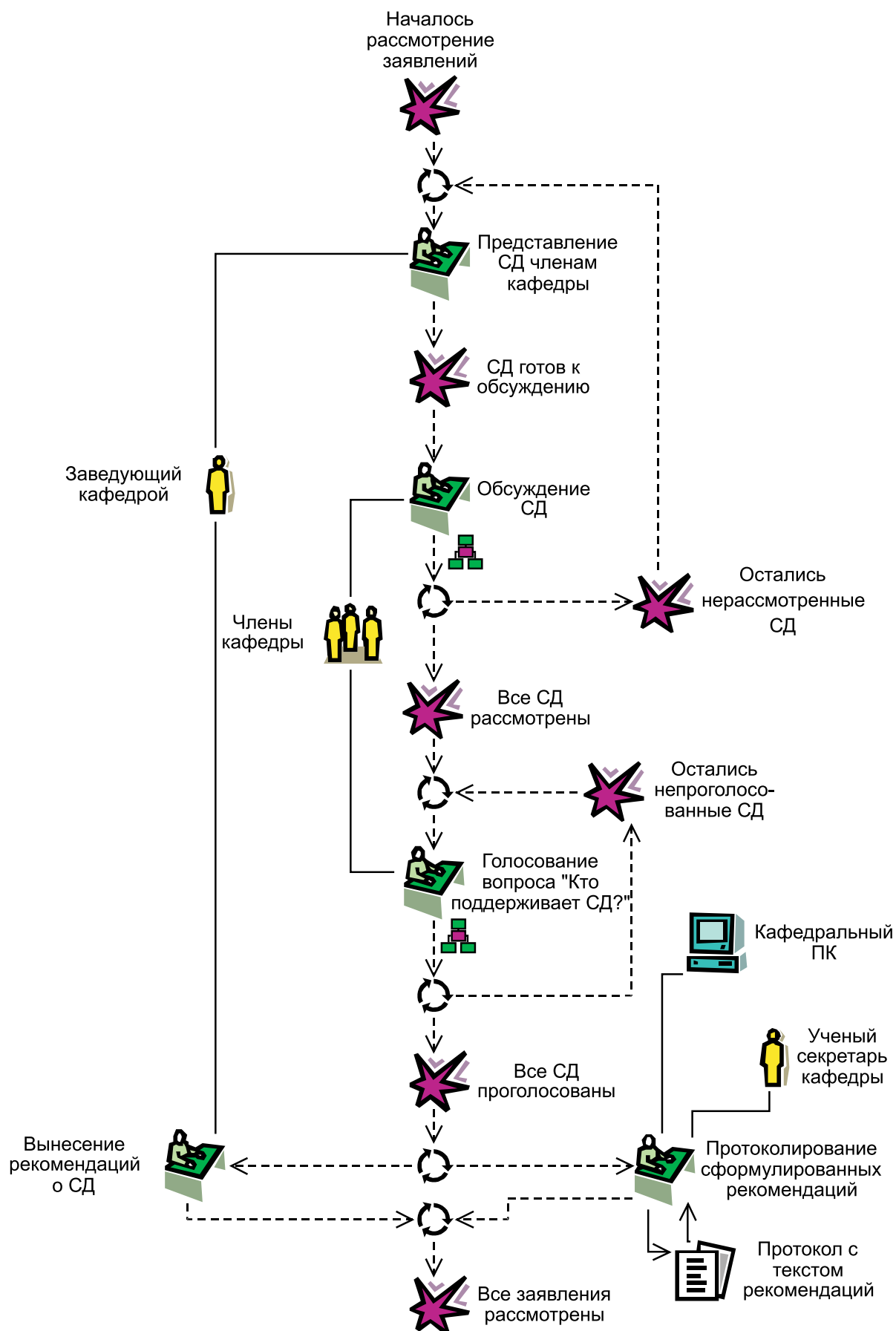


Рис.8.5.9 Функциональный поток офисного процесса  
"Рассмотрение конкурсных дел на заседании кафедры"

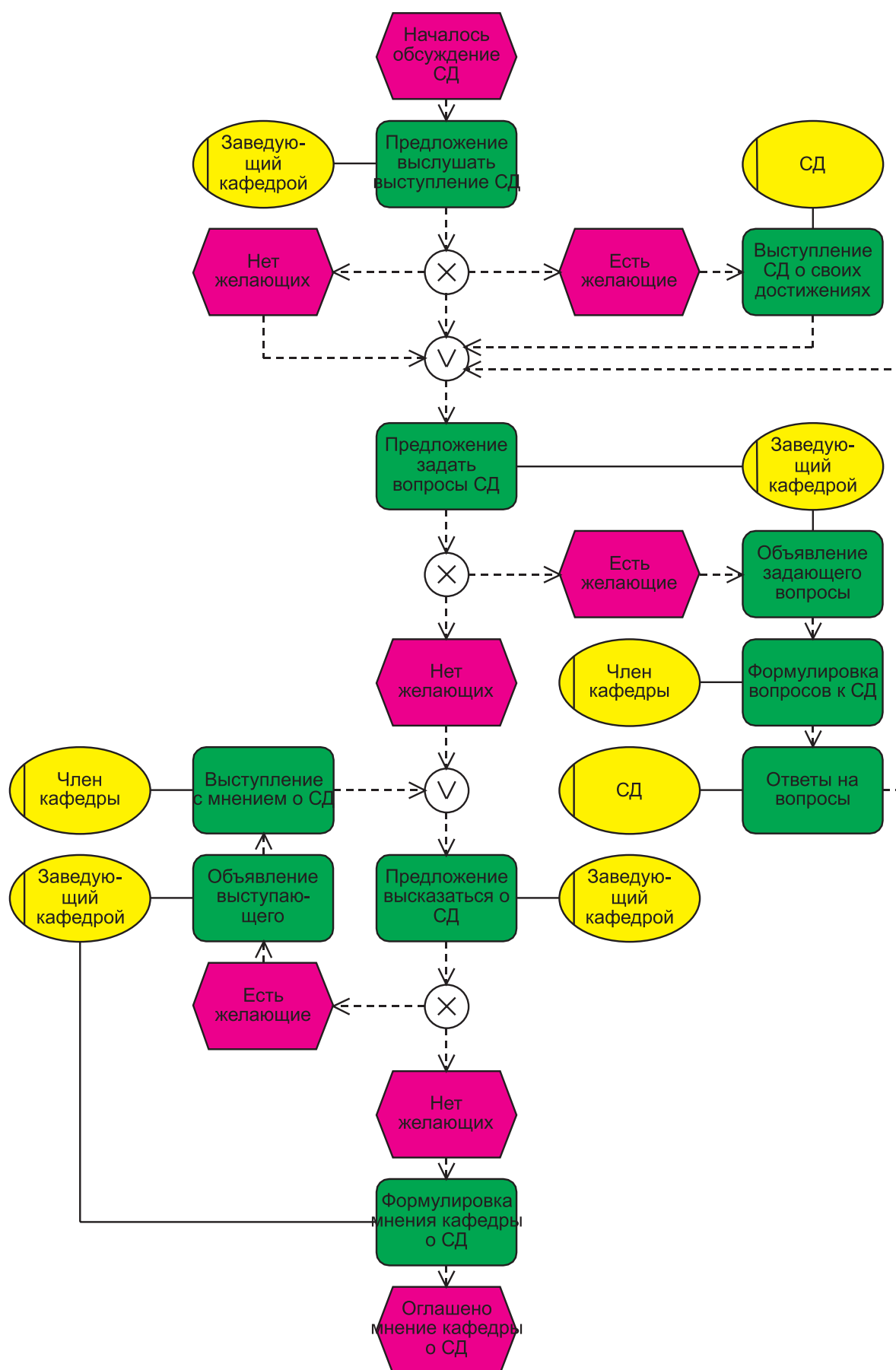


Рис.8.5.10 Функциональный поток процесса “Обсуждение СД”

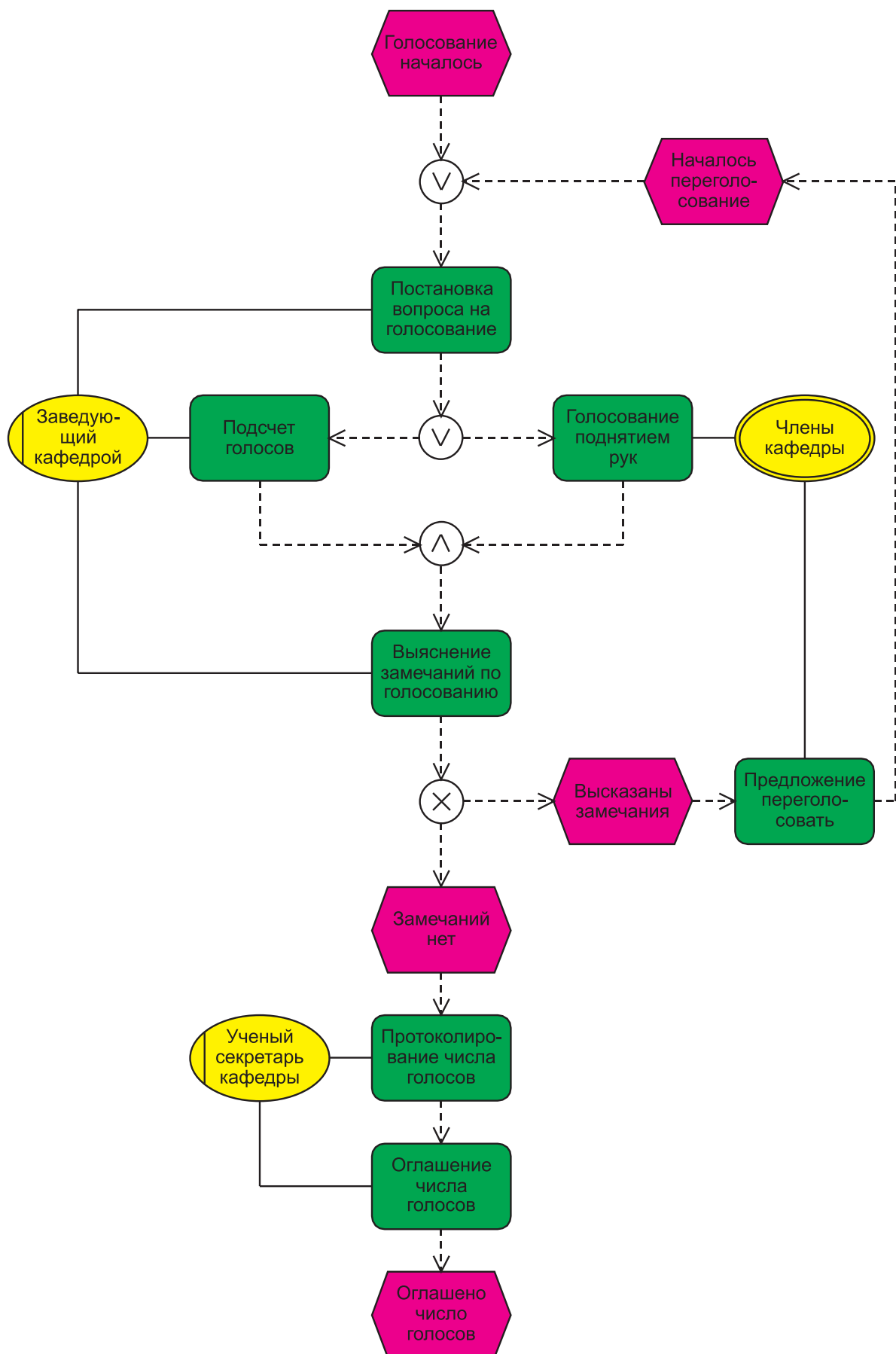


Рис.8.5.11 Функциональный поток процесса "Голосование вопроса"

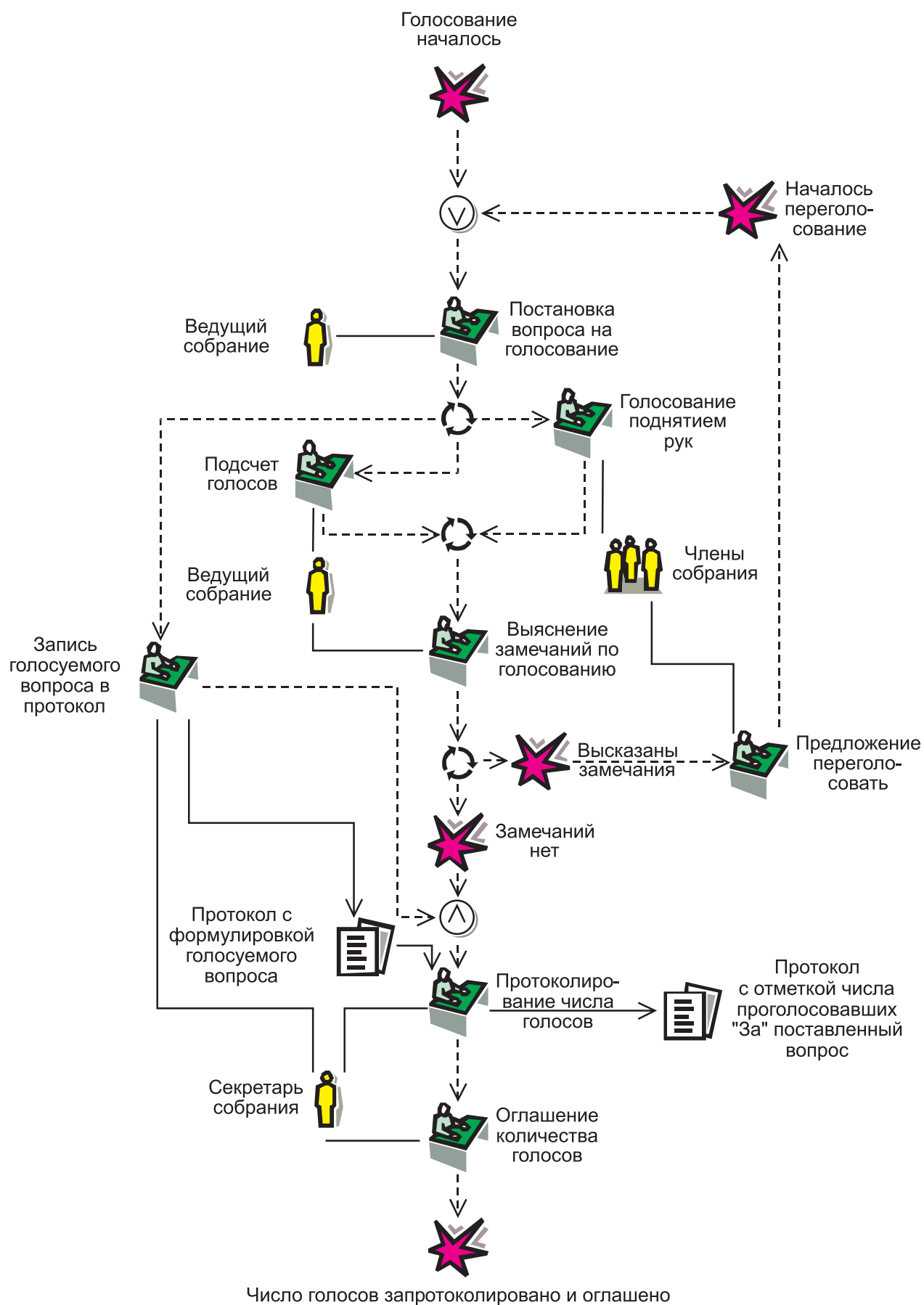


Рис.8.5.12 Функциональный поток офисного процесса "Голосование вопроса"

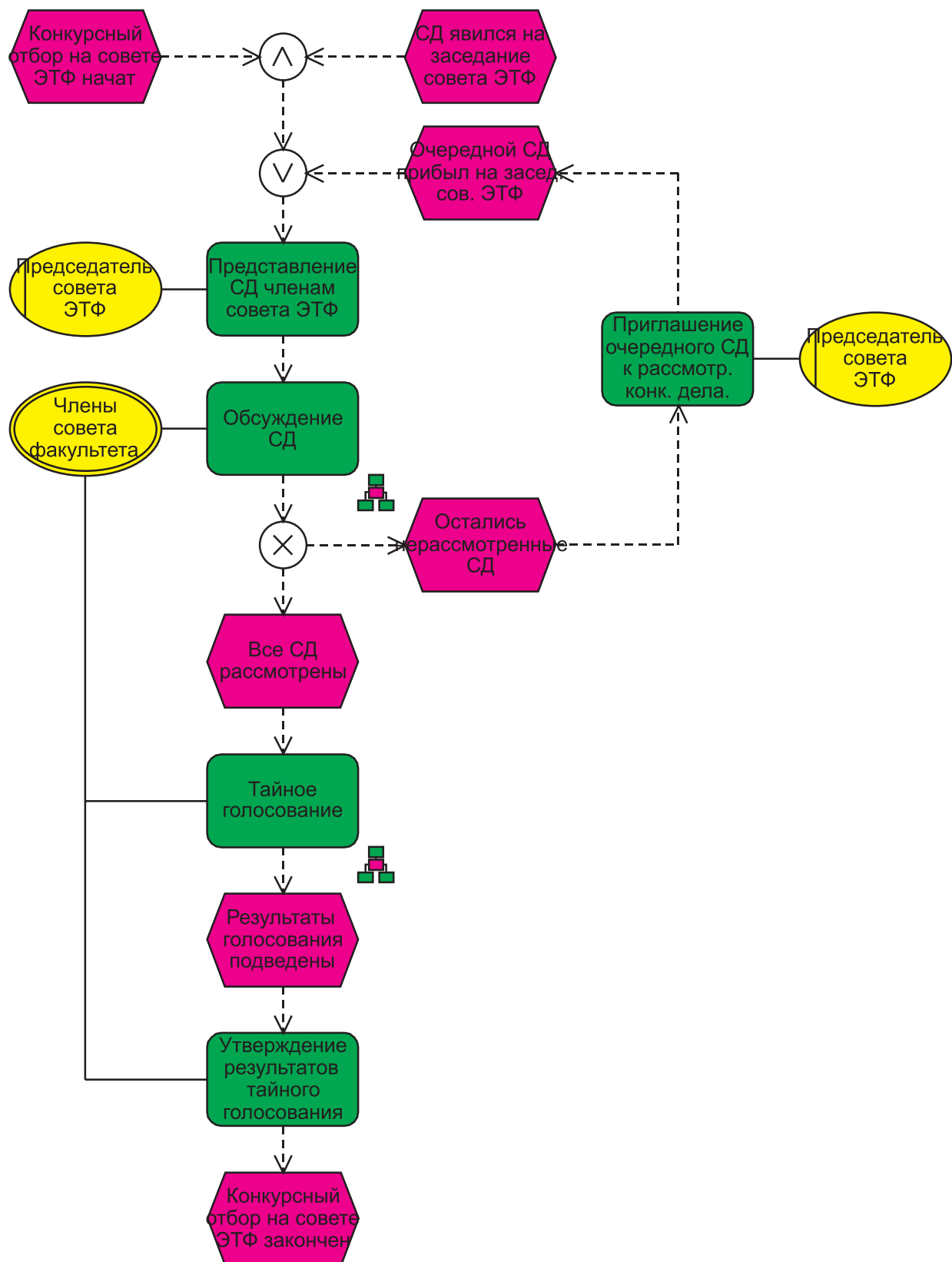


Рис.8.5.13 Функциональный поток процесса "Конкурсный отбор СД на совете ЭТФ"

снизить затраты на персонал в виде заработной платы, различного рода отчислений и т. д. При этом за счет того, что многие функции выполняются не служащими организации, а автоматизированными системами, исключаются различные простои, ошибки при выполнении тех или иных функций. Например, при переходе от офисного процесса “Голосование вопроса на собрании” к е-процессу секретарь собрания заменяется соответствующим ПО. Кроме этого, изменяется логика процесса, а также последовательность и содержание функций. Рассмотрим сценарий е-процесса “Голосование вопроса на собрании”.

Участники собрания находятся в своих офисах, дома или на базе отдыха. Они имеют при себе компьютер (Notebook), выход в Интернет (мобильные средства связи) и соответствующее ПО. Идет процесс, например, заседания виртуальной кафедры, и наступает этап принятия (большинством голосов) решения по одному из кафедральных вопросов путем голосования. Ведущий собрания создает электронный документ (ЭД) с постановкой вопроса на голосование. ПО рассылает ЭД по адресам участников собрания через электронную почту (ЭП). Участники собрания голосуют и отправляют свое мнение по ЭП. ПО в процессе голосования подсчитывает количество голосов. Затем ПО анализирует количество проголосовавших и время голосования. Если есть непроголосовавшие и осталось время для продолжения голосования, то голосование продолжается. При этом ПО делает повтор запроса на голосование по электронным адресам непроголосовавших, а если все участники собрания проголосовали, то голосование заканчивается, и ПО анализирует кворум. Если кворума нет, то ПО оповещает участников собрания о причинах срыва голосования. Если кворум есть, то ПО генерирует протокол с результатами голосования, рассылает его участникам собрания и сохраняет в электронном архиве.

На рис.8.5.14 изображён описанный е-процесс “Голосование вопроса на собрании”. Подробное описание каждой функции этого процесса можно найти в атрибутах типа Remark/Definition. В заключение приведем распечатку содержимого этих атрибутов.

**Формирование ЭД с голосуемым вопросом.** Ведущий собрание использует типовой шаблон стандартного документа с автоматически заполняемым набором определенных реквизитов: дата, время и т.д. В специально отведенном поле ведущий собрание заносит формулировку вопроса, которая ставится на голосование и по которой участники собрания должны будут проголосовать.

**Рассылка ЭД.** Сформированный в предыдущей функции ЭД рассылается по ЭП участникам собрания. При этом используются встроенные возможности ЭП, например, ведущий не сам заполняет адреса участников

собрания, а использует заранее введенный список электронных адресов участников собрания. ЭД автоматически рассылается по этим адресам. Ведущий собрание может выбрать промежуток времени, через который ЭД или соответствующий текст напоминания автоматически рассылается тем участникам собрания, от которых еще не получен результат голосования по поставленному вопросу.

**Голосование вопроса.** Получив по ЭП ЭД с голосуемым вопросом, участники собрания начинают голосовать. Они, например, нажимают клавишу 1 (да), если голосуют “за”, или 0 (нет), если голосуют “против”. Проголосовав, они тут же отправляют результат ведущему собранию по ЭП.

**Подсчёт голосов.** ПО, получив результаты голосования от участников собрания, подсчитывает количество голосов, например, путём суммирования чисел  $1+0+1+1+\dots$ .

**Анализ проголосовавших.** После того, как ПО приняло результаты от участников собрания и подсчитало количество голосов, начинается процедура анализа числа проголосовавших. ПО проверяет, правильно ли проголосовали участники собрания и все ли они ответили. Если кто-то не проголосовал, то их адреса вносятся в список непроголосовавших.

**Повтор запроса на голосование.** Если оказалось, что число проголосовавших меньше кворума, то непроголосовавшим отправляется повторный запрос на голосование. Участники собрания, которые не проголосовали, должны повторно проголосовать за определённый период времени и отправить результат ведущему собранию. Затем ПО продолжает подсчёт количества голосов и их анализ.

**Анализ кворума.** ПО анализирует количество участников собрания. Если имеется кворум, то голосование считается состоявшимся.

**Оповещение о причинах срыва голосования.** Если кворума нет, то ПО оповещает участников собрания о причинах срыва голосования и формируется соответствующий отчет.

**Генерация протокола голосования.** ПО создает протокол голосования в виде стандартного документа с набором определенных реквизитов. В данный ЭД заносится количество участников собрания и результаты голосования.

**Рассылка протокола.** ПО, сгенерировав протокол с результатами голосования, рассылает его по ЭП участникам собрания.

**Архивирование протокола.** После завершения голосования ПО направляет протокол с результатами голосования в электронный архив.

Приведенный текст описания функций рассматриваемого е-процесса достаточно подробно объясняет требования к условиям и порядку выполнения этого процесса на заседании виртуальной кафедры.



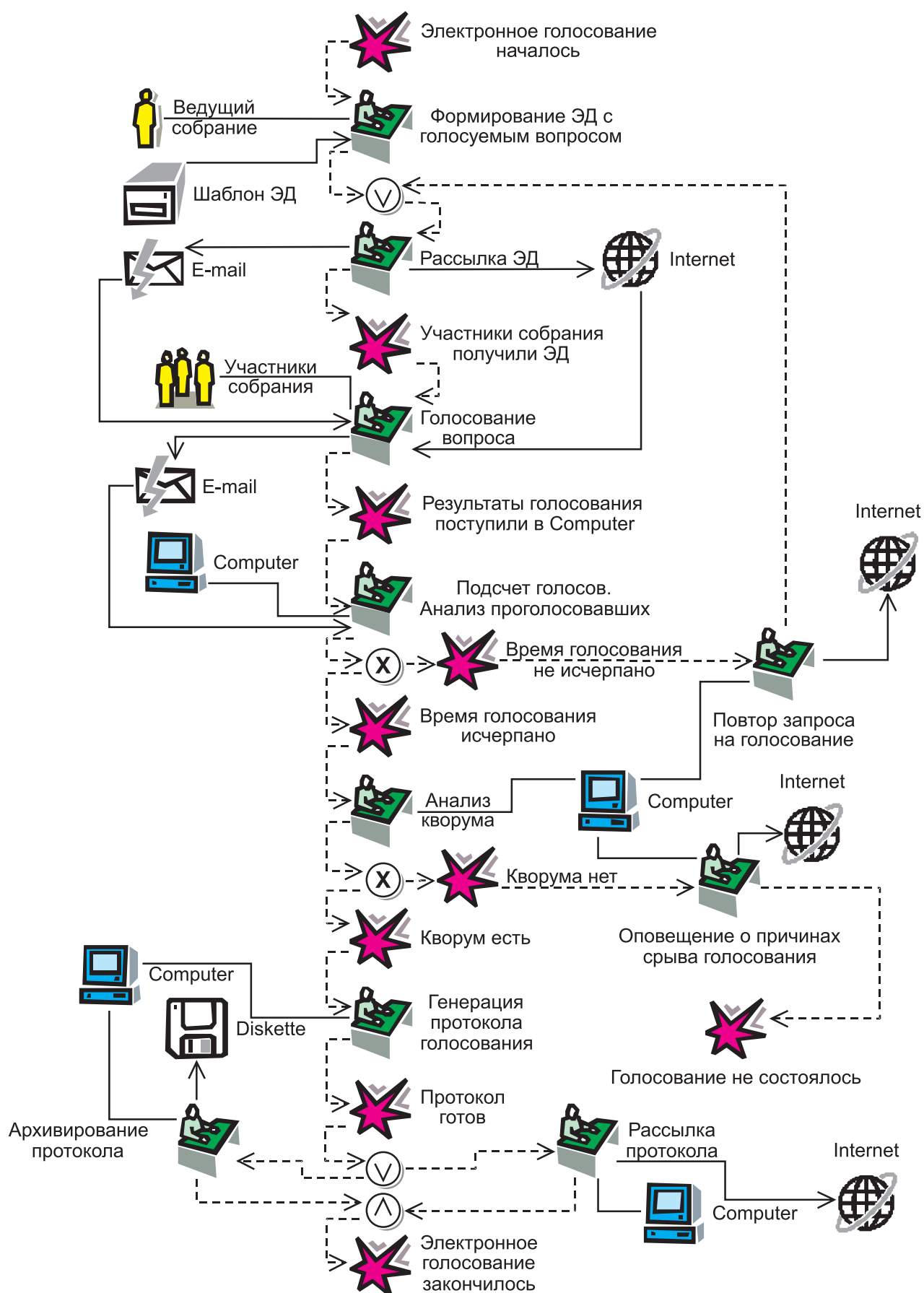


Рис.8.5.14 е-процесс "Голосование вопроса на собрании"

## 8.6. Офисная деятельность фирмы

В качестве последнего примера рассмотрим офисную деятельность малого предприятия. Это предприятие является прообразом многих известных нам малых российских фирм, которые успешно совмещают научную, производственную и коммерческую деятельность не только в своем регионе, но и далеко за его пределами.

Офис научно-производственного общества (фирмы) с ограниченной ответственностью под названием “AAA” (сокращенно ООО “AAA”) расположен в городе Челябинске. Организационная структура фирмы представлена на рис.8.6.1. Здесь каждая организационная единица выполняет определенные функции. Но этого из диаграммы не видно, так как большая часть информации является внутренней и содержится в атрибутах ARIS-объектов, например, в атрибуте типа “Описание/Определение”. Просмотреть и проанализировать эту информацию можно в отчете, созданном, например, на основе скрипта ModelObjects (рис.8.6.2).

Из организационной диаграммы и составленных по ней отчетов можно увидеть, что функции структурного подразделения “Бухгалтерия” выполняют главный бухгалтер и бухгалтер. Функции, выполняемые этим подразделением, типичны для многих организаций. Руководителем этого подразделения является главный бухгалтер, в подчинении которого находится бухгалтер. В свою очередь, главный бухгалтер несет ответственность за выполнение предписанных функций перед руководством предприятия.

Таким образом, путем анализа организационной структуры фирмы с помощью отчетов можно делать выводы о штатных должностях, количестве структурных подразделений, количестве служащих конкретного структурного подразделения фирмы, выполняемых функциях, характере разделения труда между подразделениями фирмы и т. д.

В фирме используется множество понятий (технических терминов), которые отражают ее род деятельности. На рис.8.6.3 - 8.6.11 представлены модели взаимосвязи этих понятий. Приведем определения основных понятий, используемых в ООО “AAA”, которые сформированы из содержимого атрибутов типа Description/Definition представленных моделей.

**Контрагент** – юридическое или физическое лицо, берущее на себя известные обязательства по договору.

**Поставщик** – юридическое или физическое лицо, источник приобретения материальных ценностей и/или получения услуг.

**Потенциальный поставщик** – поставщик, являющийся потенциальным источником приобретения материальных ценностей и/или получения услуг.

**Постоянный поставщик** – поставщик, являющийся источником приобретения материальных ценностей и/или получения услуг на постоянной договорной основе.

**Покупатель** – юридическое или физическое лицо, которому реализуются материальные ценности и/или оказываются услуги.

**Потенциальный покупатель** – покупатель, имеющий потребность в приобретении товара (работы или услуги).

**Постоянный покупатель** – покупатель, которому реализуются материальные ценности и/или оказываются услуги на постоянной договорной основе.

**Получатель** – юридическое или физическое лицо, получающее платеж по финансовому сопроводительному документу.

**Плательщик** – юридическое или физическое лицо, осуществляющее платеж по финансово-сопроводительному документу.

**Посредник** – юридическое или физическое лицо, содействующее обращению товаров и услуг на внутренних и внешних рынках.

**Принципал** – лицо, участвующее в сделке за свой счет, или лицо, от имени которого действует агент (представитель).

**Коммерческие представители** – посредники, выполняющие поручения предпринимателей (принципалов) по поиску для них коммерческих партнеров и действующие от имени принципалов при заключении ими договоров.

**Комиссионеры** – посредники, содействующие по поручениям предпринимателей (комитентов) сбыту или приобретению товаров в рамках комиссионных договоров путем заключения договоров от своего имени и за счет доверителей.

**Поверенные** – посредники, которые по поручениям предпринимателей (доверителей) содействуют совершению коммерческих операций, в том числе по сбыту или приобретению товаров путем заключения договоров от своего имени и за счет доверителей.

**Агенты** – посредники, совершающие по поручению предпринимателя (принципала) юридические и иные действия, в том числе по сбыту и приобретению товаров, путем заключения договоров от своего имени, но за счет принципала либо от имени и за счет принципала.

**Дистрибьютеры** – оптовые торговцы, которые в рамках дистрибьютерских договоров сбывают товары продавцов, заключая контракты купли-продажи.

**Документ** – деловая бумага, подтверждающая какой-нибудь факт или право на что-либо.

**Сопроводительный документ** – документ, сопровождающий движение товаров (услуг) и денежных средств. Как правило, является вторичным и может

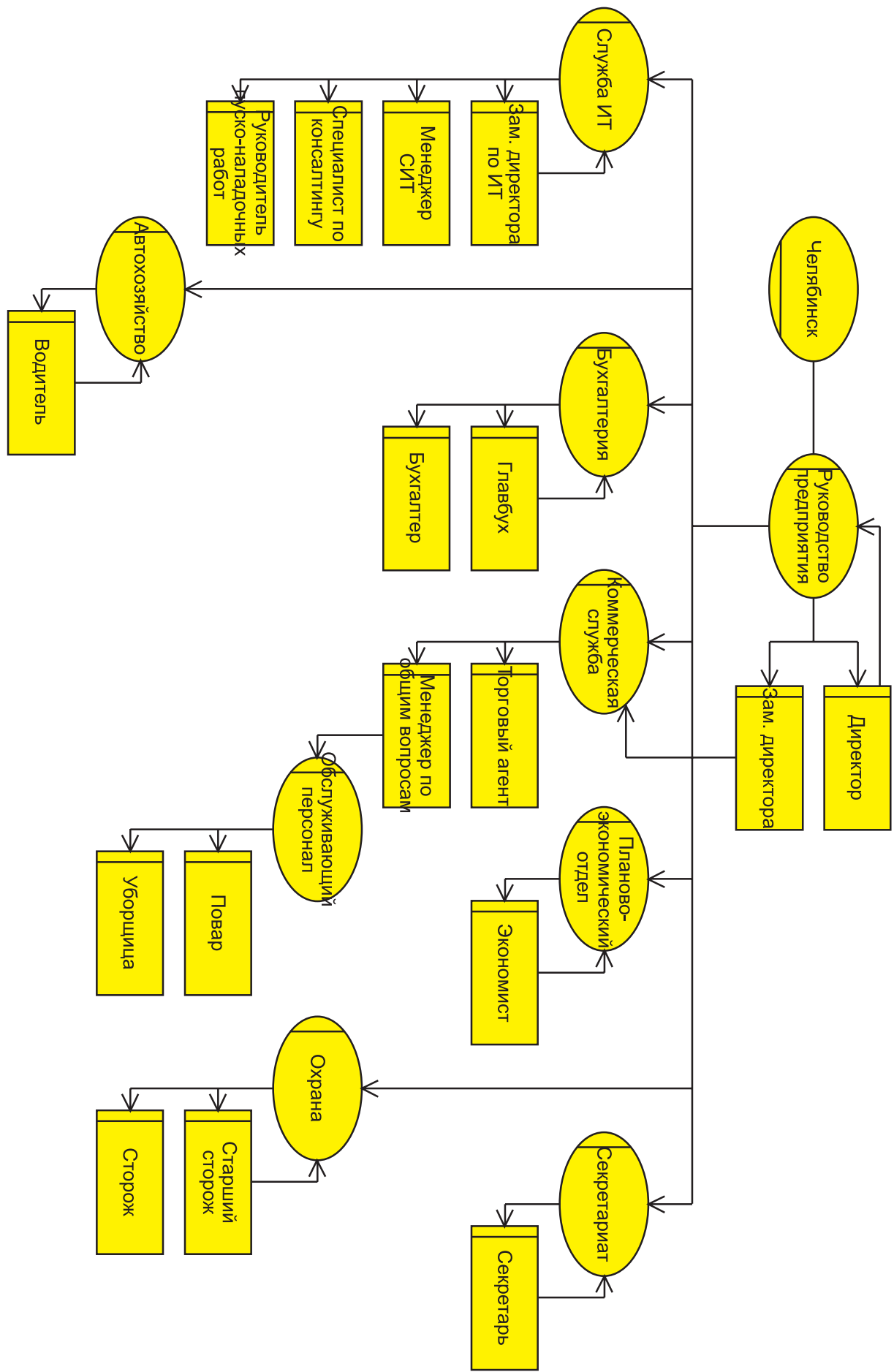




Рис.8.6.1 Организационная структура фирмы

|   |                       |   |
|---|-----------------------|---|
|  | Objects of the models |  |
|---|-----------------------|---|

## ARIS Report

Server: LOCAL

Database: AAA

User: system

|   |   |                  |
|---|---|------------------|
| <b>Organizational chart: ОргСтруктура</b> |   |                  |
| <b>Object name:</b>                       | <b>Object type:</b>   | <b>Group:</b>    |
| <b>Автохозяйство</b>                      | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Выполнение поручений руководителя организации, связанные с транспортировкой.  |                  |
| Creator:                                  | System  |                  |
| <b>Бухгалтерия</b>                        | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Ведение бухгалтерского учета, предоставление всей необходимой отчетности в налоговые органы, формирование всех сопроводительных документов.                       |                  |
| Creator:                                  | System  |                  |
| <b>Коммерческая служба</b>                | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Отвечает за подготовку и заключение договоров, обеспечение выполнения условий заключенных договоров, контроль за своевременным исполнением заключенных договоров. |                  |
| Creator:                                  | System  |                  |
| <b>Обслуживающий персонал</b>             | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Содействие выполнению основных функций других подразделений фирмы.  |                  |
| Creator:                                  | System  |                  |
| <b>Охрана</b>                             | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Несет ответственность за сохранность имущества фирмы.   |                  |
| Creator:                                  | System  |                  |
| <b>Планово экономический отдел</b>        | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Расчет рентабельности сделок, разработка схем и бюджетов сделок, составление других экономических расчетов.   |                  |
| Creator:                                  | System  |                  |
| <b>Руководство предприятия</b>            | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Управление финансово-хозяйственной деятельностью фирмы, разработка стратегии развития и принятие решений.   |                  |
| Creator:                                  | System  |                  |
| <b>Секретариат</b>                        | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Своевременное и точное выполнение поручений руководителя фирмы, фильтрация информации.  |                  |
| Creator:                                  | System  |                  |
| <b>Служба ИТ</b>                          | <b>Organizational unit</b>  | <b>ОргМодели</b> |
| Description/Definition:                   | Работа с новейшими ИТ, внедрение их в процесс производства и оказания услуг.  |                  |

Рис.8.6.2 Отчет по организационной диаграмме на основе скрипта ModelObjects

быть связан с определенным документом-основанием.

*Организационно-распорядительные документы* – документы, в которых фиксируются вопросы управления, взаимодействия, обеспечения и регулирования деятельности органов власти и управления фирмы, ее подразделений и должностных лиц.

*Документ-основание* – операционный документ, регламентирующий операции с другими юридическими лицами.

*Входящий документ* – документ, поступивший на фирму от сторонней организации.

*Исходящий документ* – документ, созданный на фирме и отправленный адресату, копия которого, как правило, остается в деле.

*Оригинал документа* – документ, атрибуты которого являются оригинальными.

*Копия документа* – ксерокопия или факс-копия оригинала документа.

*Проект документа* – предварительный, неподписанный вариант документа, предназначенный для согласования.

*Оформленный документ* – документ, согласованный на всех уровнях и утвержденный директором фирмы.

*Согласованный документ* – обсужденный и оцененный всеми заинтересованными должностными лицами проект документа.

*Письмо* – обобщенное название различных по содержанию документов, объединенных едиными правилами составления и оформления.

*Сопроводительное письмо* – документ, информирующий адресата о прилагаемых к письму документах.

*Гарантийное письмо* – документ, гарантирующий исполнение изложенных в документе обязательств.

*Письмо на оплату* – документ, указывающий сумму и реквизиты оплаты.

*Письмо с реквизитами отгрузки* – документ, указывающий реквизиты отгрузки товара.

*Письмо-поручение* – документ с указанием действий, которые должен совершить агент, и условий совершения этих действий.

*Отчет* – документ, содержащий сведения о выполнении задания, работы, подготовке мероприятий, поручений и проведении тех или иных мероприятий, представляемый вышестоящему учреждению, подразделению фирмы или должностному лицу.

*Договор* – документ, фиксирующий соглашение сторон об установлении каких-либо отношений и регулирующий эти отношения.

*Коммерческое предложение* – выраженное в устной или письменной форме предложение товара (услуги) потенциального поставщика потенциальному покупателю

с указанием условий возможного сотрудничества.

*Информационная система* – совокупность аппаратных и программных средств, обеспечивающая ввод, редактирование и хранение информации о финансово-хозяйственной деятельности фирмы.

*Информационная технология* – совокупность методов, устройств и производственных процессов, используемых людьми для сбора, хранения, обработки и распространения информации.

ООО «ААА» осуществляет свою деятельность в научно-производственном и коммерческом направлениях. Миссией фирмы является сертифицированная поддержка предприятий заказчиков освоенными информационными технологиями (ИТ) ведения бизнеса и управления персоналом. ИТ внедряются не только на предприятиях-клиентах, но, в первую очередь, на самой фирме. Площадкой для такого внедрения является активно функционирующее коммерческое направление, где освоенные ИТ апробируются в реальных условиях и используются всеми участниками коммерческой деятельности фирмы.

Научно-производственную деятельность фирмы описывает следующая последовательность основных БП: анализ рынка и отбор ИТ, освоение ИТ, работа с клиентами. Модель этих процессов и их составляющих подпроцессов описана в диаграмме на рис.8.6.12. Рассмотрим подробное описание этих процессов.

В процессе анализа рынка ИТ можно выделить следующие подпроцессы: сбор информации о рынке ИТ, анализ доступных ИТ, отбор ИТ к освоению. Рассмотрим подробности только первого из перечисленных подпроцессов, т. е. сбора информации о рынке ИТ.

Для сбора информации о рынке ИТ специалисты службы ИТ (СИТ) используют интернет, периодические издания (газеты, журналы и т. д.), рекламу на телевидении и радио, специализированные выставки, презентации и семинары. Интернет используется на рабочем месте секретаря и заместителя директора по ИТ (ЗДИТ). Секретарь использует интернет только для получения электронной почты (в соответствии со своими должностными обязанностями), в содержании которой также может быть информация о новых ИТ. ЗДИТ использует интернет непосредственно для получения информации о состоянии дел на рынке ИТ, а также для переписки с производителями и распространителями ИТ по вопросам сотрудничества.

Таким образом, процесс получения через интернет информации о состоянии дел на рынке ИТ можно разбить на подпроцесс получения информации через секретаря и подпроцесс получения информации через ЗДИТ. Первый подпроцесс протекает по следующему сценарию.

Секретарь активизирует интернет-соединение три ра-

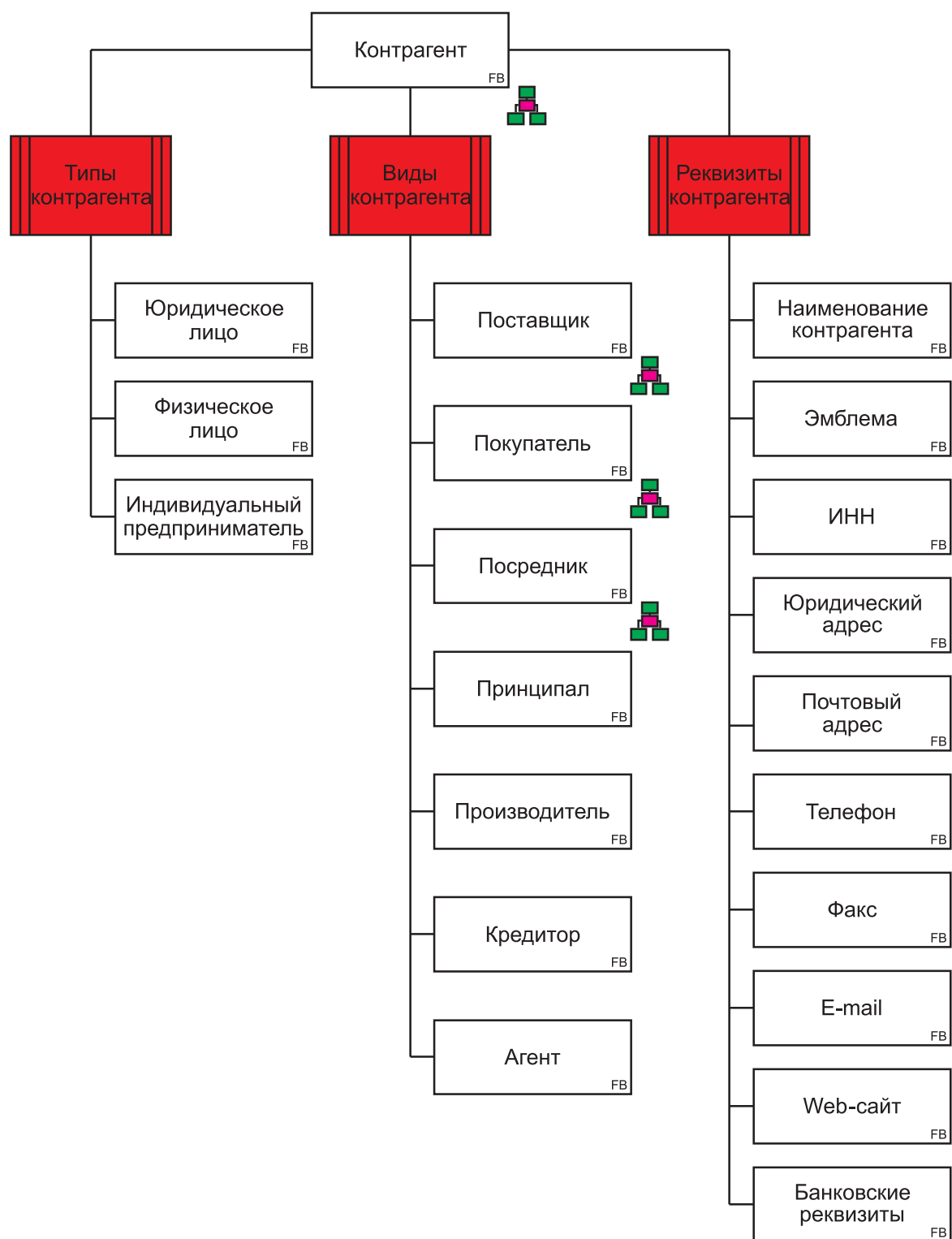


Рис.8.6.3 Модель технических терминов “Контрагент”

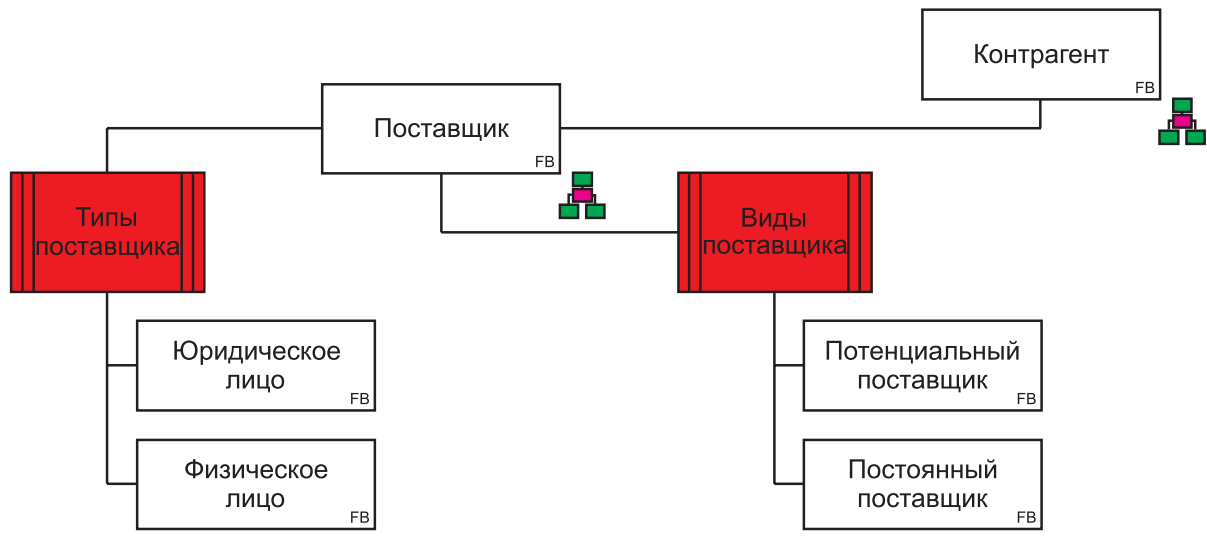


Рис.8.6.4 Модель технических терминов “Поставщик”

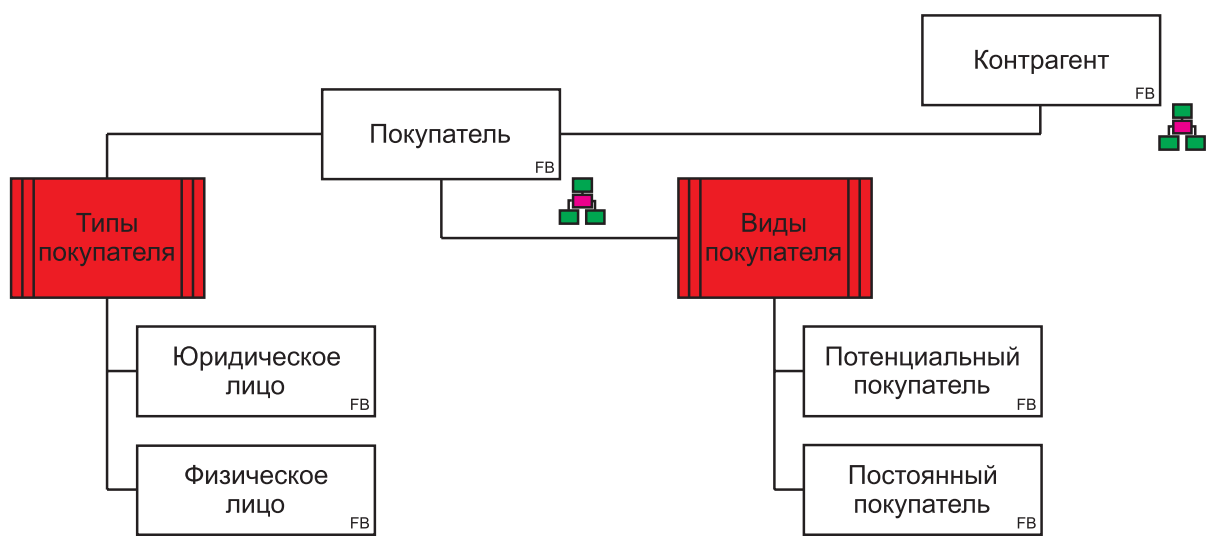


Рис.8.6.5 Модель технических терминов “Покупатель”

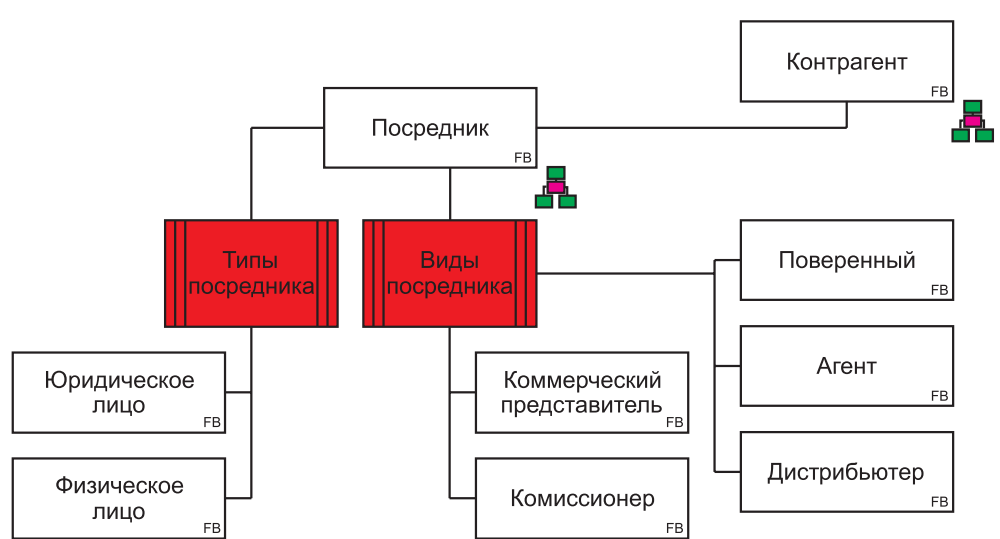


Рис.8.6.6 Модель технических терминов “Посредник”



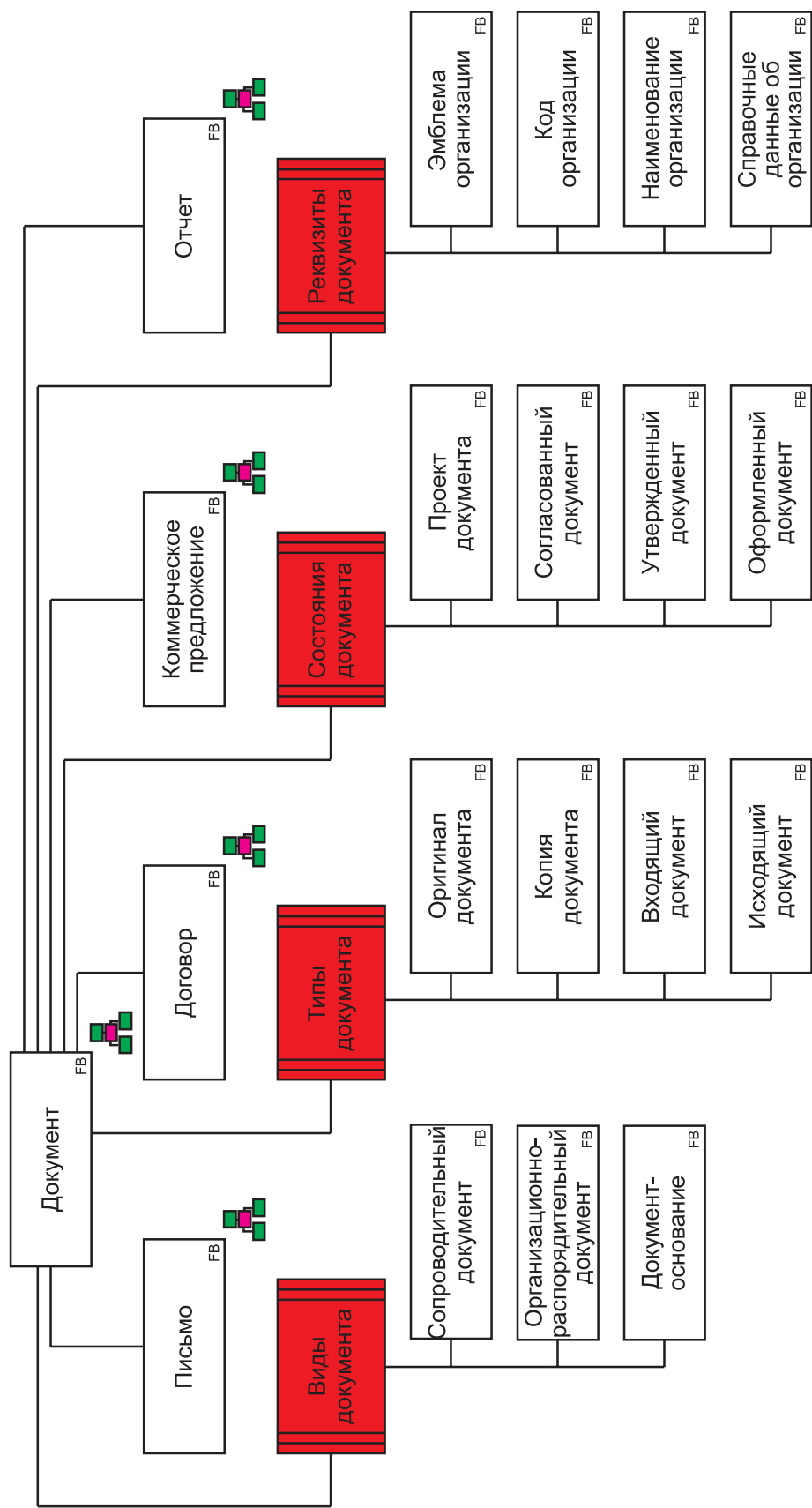


Рис.8.6.7 Модель технических терминов “Документ”

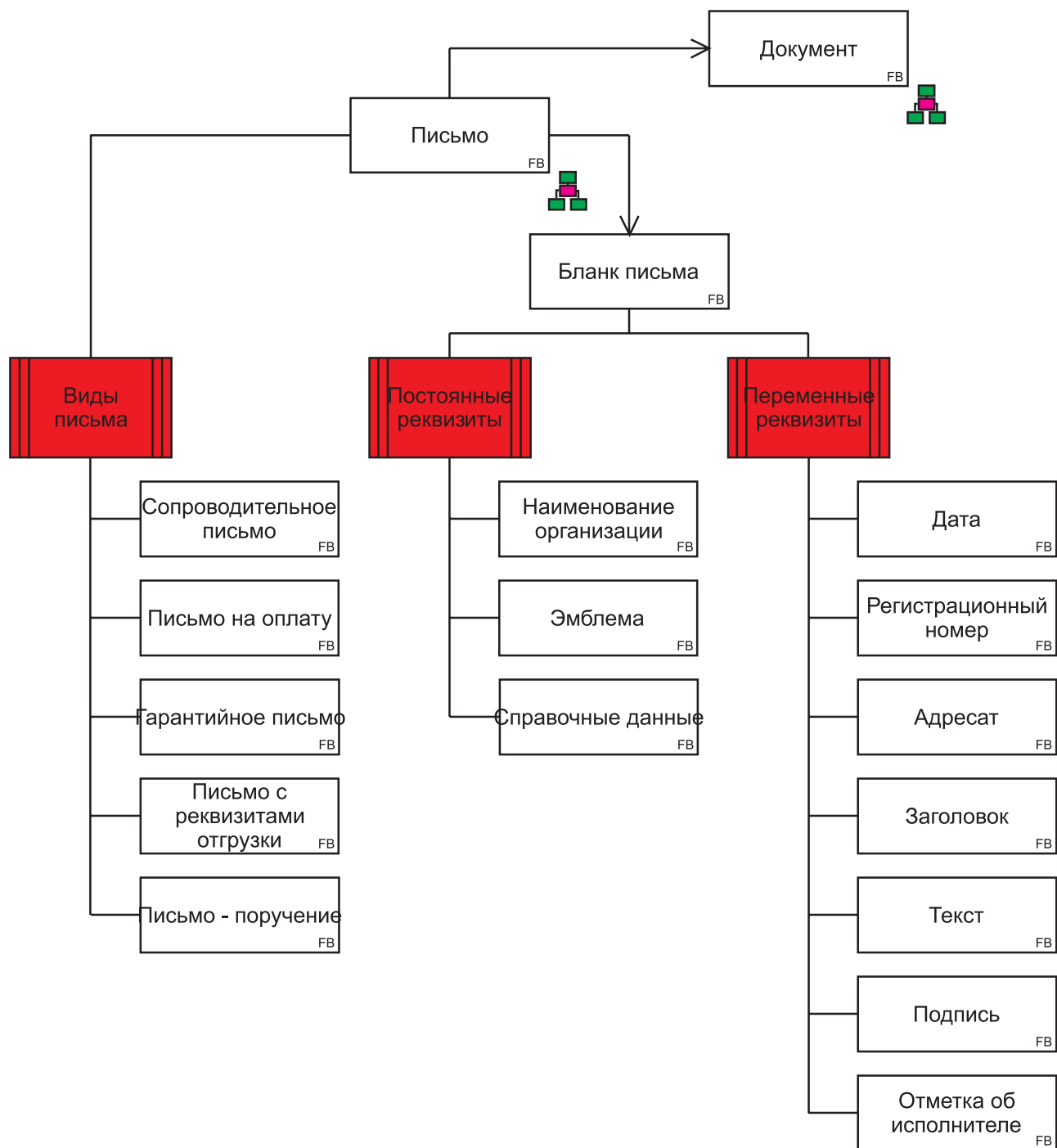


Рис.8.6.8 Модель технических терминов "Письмо"

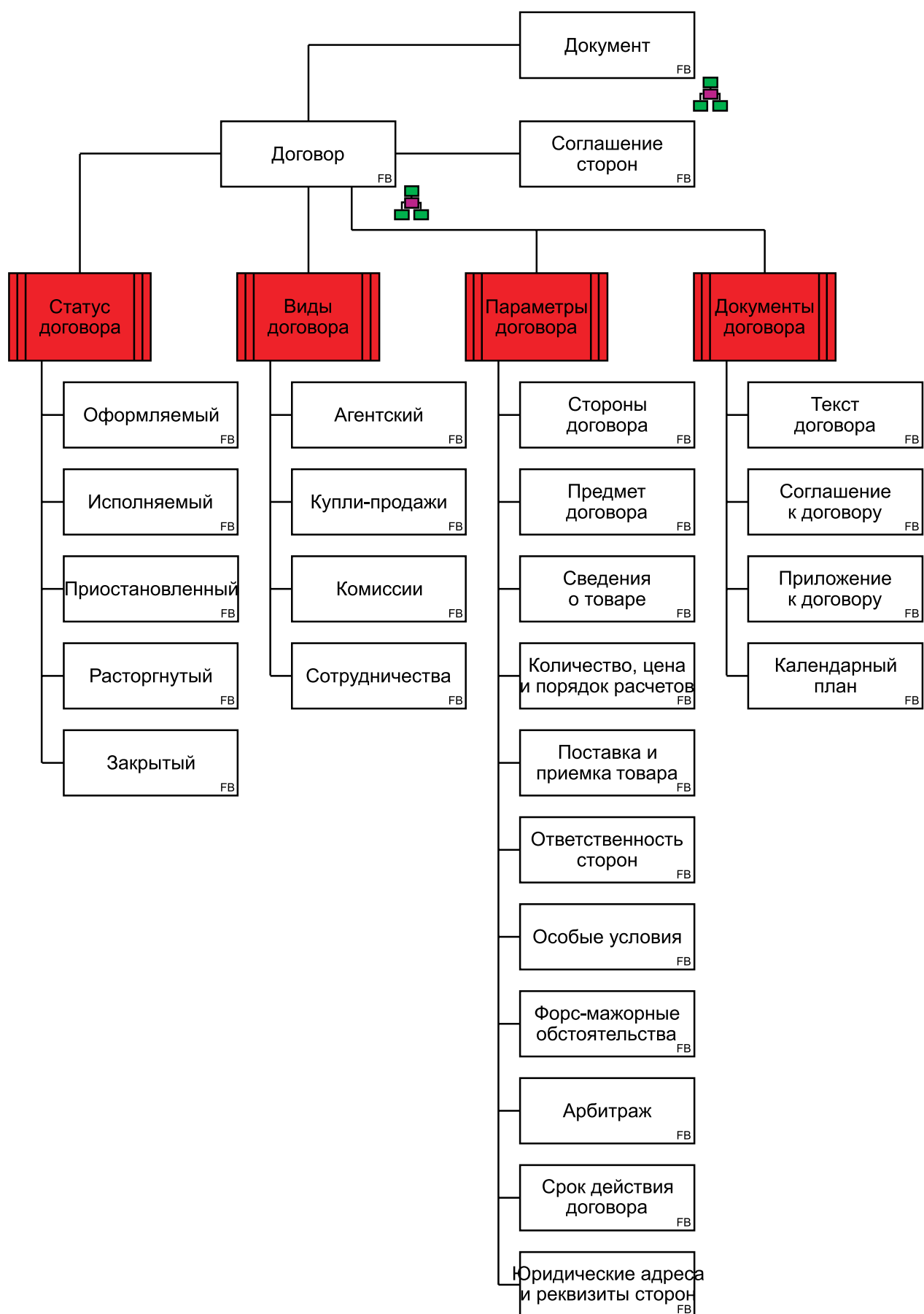


Рис.8.6.9 Модель технических терминов “Договор”

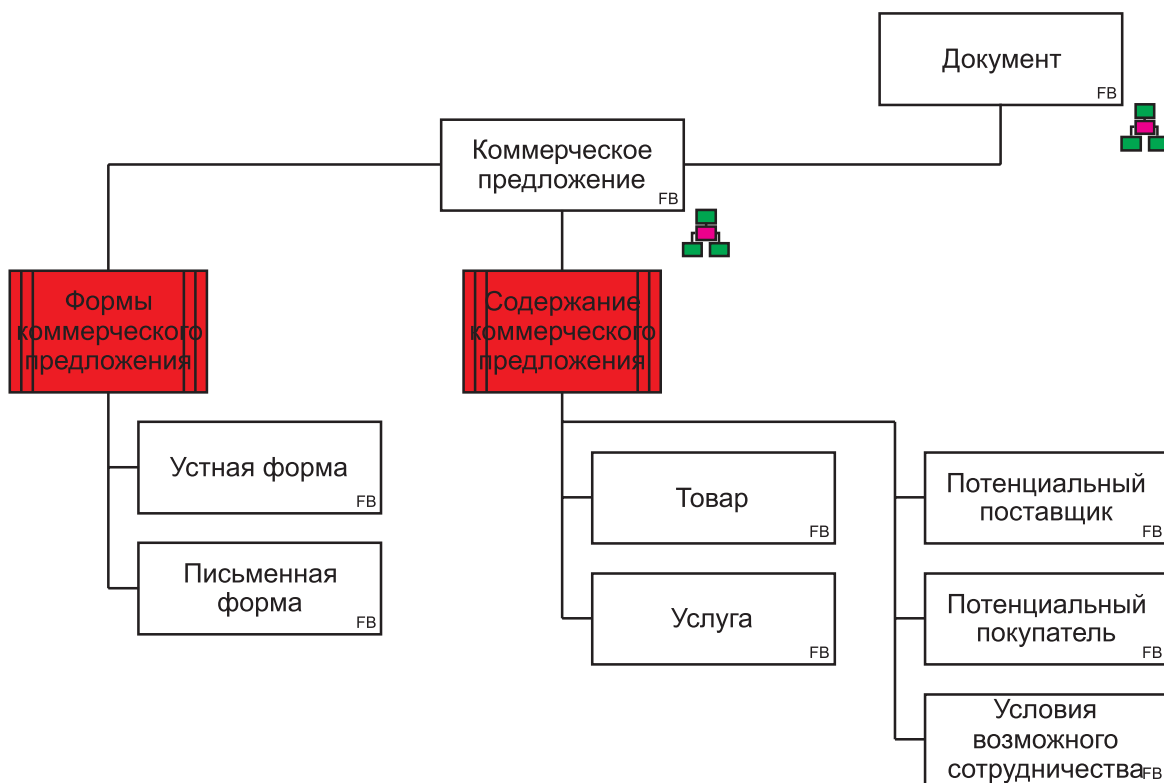


Рис.8.6.10 Модель технических терминов "Коммерческое предложение"

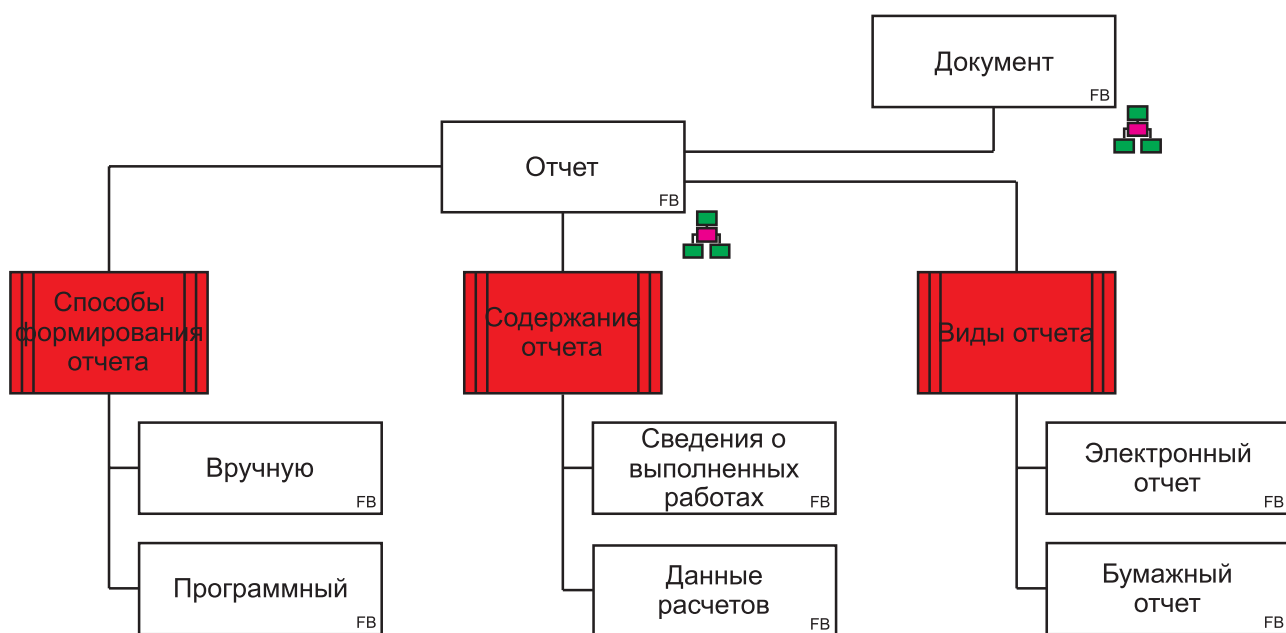


Рис.8.6.11 Модель технических терминов "Отчет"

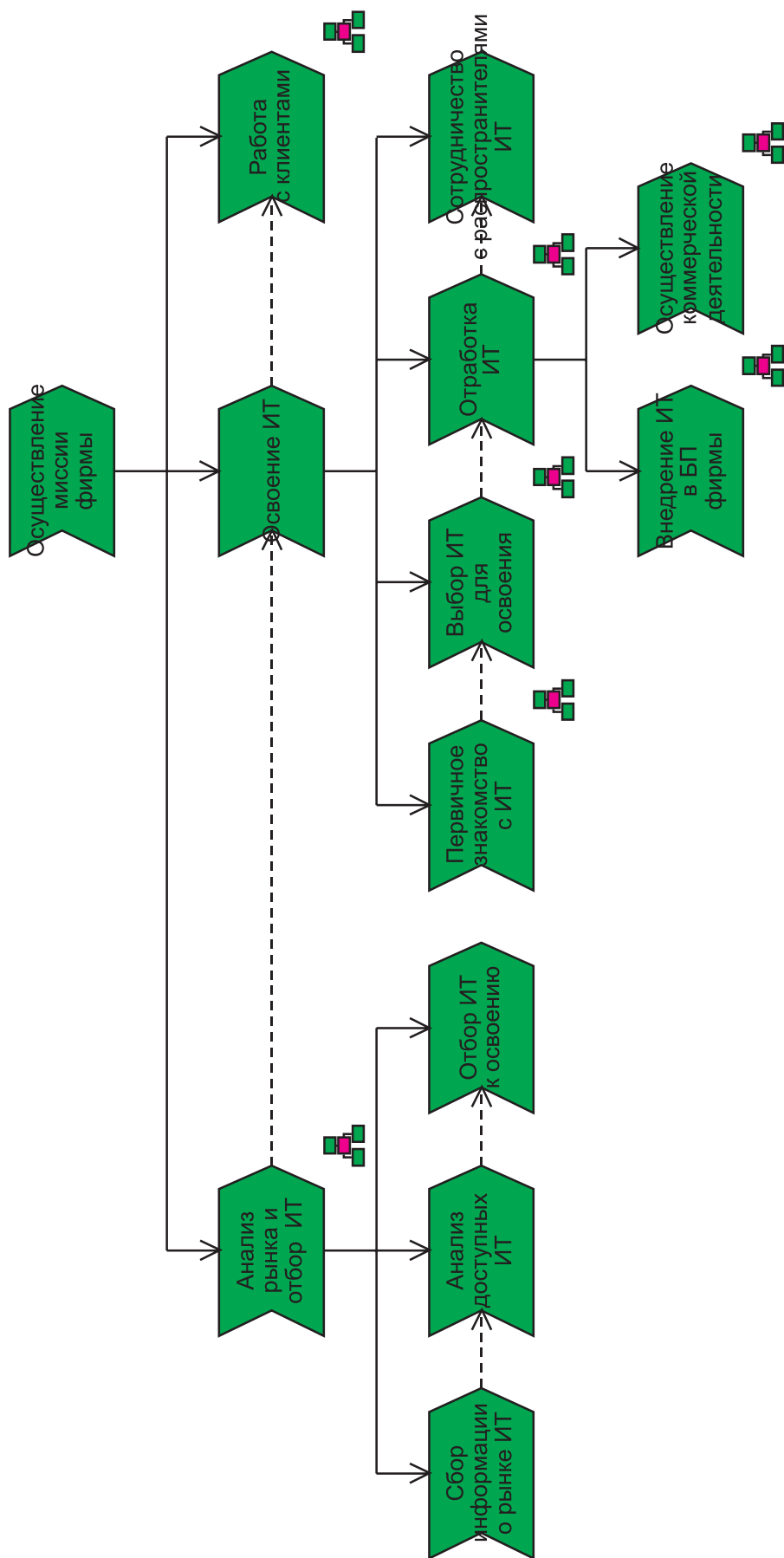


Рис.8.6.12 Модель основных процессов ООО “ААА”

за в день. Он может производить дополнительную активизацию, если знает, что должно быть послано или получено электронное сообщение. После активизации соединения осуществляется прием электронных сообщений. После этого секретарь производит рассылку полученной информации по внутренней электронной почте. Информация рассылается сотрудникам, рабочие места которых оснащены компьютерами и настроены на использование электронной почты. Таким образом, вся информация, получаемая посредством электронной почты, доступна сотрудникам фирмы для ознакомления.

Получение информации через ЗДИТ протекает по следующему сценарию.

ЗДИТ активизирует интернет-соединение и принимает электронные сообщения по мере необходимости и выполнения своих функциональных обязанностей. Часть сообщений пересылается секретарю для дальнейшей рассылки сотрудникам. После получения информации секретарь рассылает ее сотрудникам по электронной почте.

Для сбора информации о состоянии дел на рынке ИТ фирма выписывает периодические издания, например, журналы “PC Week RE”, “ComputerWorld RE”, “Экономика и Жизнь”, “Галактика” и др.

Ответственное лицо получает почту и передает ее секретарю. Секретарь производит распределение полученных изданий между отделами и службами в соответствии с назначением изданий. После поступления последних в СИТ ее начальник (ЗДИТ) назначает время, в течение которого происходит чтение полученных изданий и отбор наиболее интересных статей. Далее ЗДИТ может заострить внимание того или иного сотрудника фирмы (менеджера СИТ, директора, главного бухгалтера, экономиста) на отработанном материале.

Остановимся подробнее на сборе информации о рынке ИТ путем участия в выставках (презентациях, семинарах).

Вопрос о посещении специализированной выставки встает в случае получения приглашения от организаторов или заинтересованности ЗДИТ. Последний сообщает о будущем мероприятии директору и подает запрос на его посещение с указанием места проведения и состава участников. Директором и его заместителем принимается решение удовлетворить или не удовлетворить запрос службы СИТ. Их решение основывается на данных о финансовом положении фирмы, предоставленных экономистом, предполагаемых затратах на посещение выставки и доводах специалистов СИТ в пользу посещения данной выставки. Обоснованный отказ в посещении выставки передается ЗДИТ. Последний вправе уточнить причины отказа и попытаться привести дополнительные доводы в пользу удовлетворения запроса. В этом случае

запрос может быть повторно рассмотрен руководством, и решение может быть пересмотрено.

В случае принятия решения об удовлетворении запроса о посещении специализированной выставки последний утверждается и передается заместителю директора. Он, в свою очередь, оповещает об этом заинтересованных лиц и при необходимости уточняет информацию о мероприятии. Далее поручает специалисту СИТ или самостоятельно проводит организационные мероприятия по подготовке к посещению выставки (готовит заявку на автотранспорт, резервирует денежные средства и т. п.).

В назначенное время происходит запланированное посещение специализированной выставки, в ходе которого специалисты получают необходимую информацию о представленных там ИТ. По возможности они получают демонстрационные материалы и готовят почву для потенциального сотрудничества с распространителями ИТ.

После посещения выставки ЗДИТ (или ответственный специалист СИТ) представляют отчет о мероприятии, т. е. доводят информацию, полученную в ходе посещения выставки, до руководства фирмы (на представлении отчета помимо руководства могут присутствовать также экономист и главный бухгалтер).

Перейдем к описанию процесса освоения ИТ. Этот процесс (смотрите рис.8.6.1) состоит из следующих подпроцессов: первичное знакомство с ИТ, выбор ИТ для освоения, отработка ИТ, сотрудничество с распространителями ИТ. На рис.8.6.13 изображена eEPC-диаграмма первичного знакомства с ИТ.

Если на этапе получения информации о состоянии рынка ИТ были получены демонстрационные материалы (демо-материалы), то ответственный за освоение данной ИТ приступает к ее первичному изучению. Если демо-материалы получить не удалось, то действия для их получения проводятся по следующему сценарию.

Специалисты СИТ или ЗДИТ посредством телефонного звонка или электронной почты уточняют информацию по вопросу получения или приобретения демонстрационных материалов. Полученная информация передается директору, и происходит ее обсуждение с целью принятия решения о том, приобретать (получать) демонстрационные материалы на предложенных условиях или нет.

Если принято решение о приобретении материалов, то специалисты СИТ или ЗДИТ посредством телефонного звонка или электронной почты связываются с распространителями рассматриваемой ИТ и формируют заказ на ее демонстрационные материалы. Если последние распространяются на платной основе, то выясняются возможные формы оплаты. Эта информация передается директору, его заместителю и главному бухгалтеру, кото-

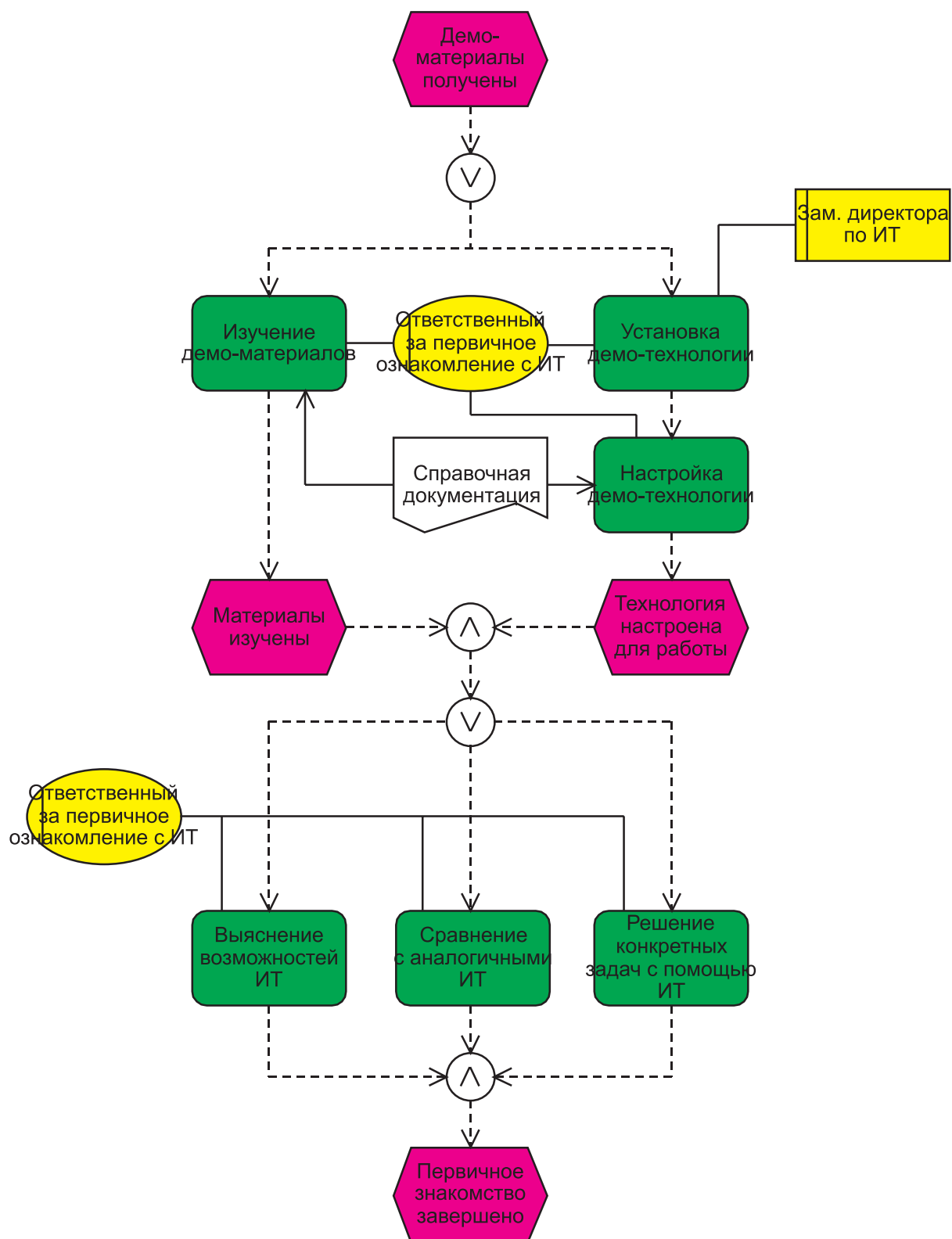


Рис.8.6.13 Первичное знакомство с ИТ



рые принимают решение о том, каким образом оплатить приобретение демонстрационных материалов.

Если принято решение оплачивать безналичным расчетом, то специалисты СИТ или ЗДИТ посредством телефонного звонка или электронной почты связываются с распространителями рассматриваемой ИТ и общаются об этом решении. Далее ожидается получение счета на оплату от контрагента. Последний присылает счет на оплату по факсу. Секретарь его принимает и передает по соответствующему маршруту. Далее директором принимается решение об оплате выставленного счета на основании данных о финансовом состоянии фирмы, предоставленных планово-экономическим отделом и консультации с главным бухгалтером. Если принято решение оплачивать счет, то он вводится в ИС специалистом СИТ и производится его оплата.

Процесс оплаты зависит от того, в каком режиме (обычном или в агентском) в данный момент функционирует фирма. Если фирма функционирует в агентском режиме, то оплата счета осуществляется следующим образом.

Заместитель директора поручает экономисту подготовить письмо Принциалу об оплате. Экономист готовит соответствующее письмо, предварительно уточнив информацию о платежных реквизитах контрагента. Подготовленное письмо передается для согласования главному бухгалтеру, который проверяет оформление и содержание письма, при необходимости высказывает свои замечания, визирует его и возвращает экономисту, который передает его на утверждение директору.

Директор просматривает и согласовывает со специалистами полученный документ. Затем подписывает его и передает секретарю, который проставляет печать и отправляет письмо Принциалу по факсу. После этого ожидается оплата от Принциала.

Как только Принциал перечисляет денежные средства по реквизитам, указанным в письме, он присылает по факсу копию платежного поручения, которую принимает секретарь и пересылает по соответствующему маршруту. Факт оплаты фиксируется в ИС экономистом после получения им копии платежного поручения (в бумажном или электронном варианте).

ЗДИТ поручает секретарю отправить копию платежного поручения контрагенту с целью уведомить его о факте произведенной оплаты.

Если фирма функционирует в обычном режиме, то оплата счета производится следующим образом.

Заместитель директора поручает бухгалтеру подготовить платежное поручение к выставленному счету. Бухгалтер готовит платежное поручение с помощью ИС и лично передает его для согласования главному бухгалтеру. По-

следний проверяет содержание и оформление документа и, если все верно, передает его обратно бухгалтеру. После согласования платежного поручения с главным бухгалтером бухгалтер лично передает его директору на утверждение. Утвержденный директором документ передается секретарю, который проставляет печать и возвращает его бухгалтеру. Бухгалтер делает заявку на автомобиль. В определенное время бухгалтер, используя автотранспорт фирмы (или в некоторых случаях общественный транспорт) отвозит платежное поручение в банк, который в случае правильного оформления принимает его к исполнению, т. е. переводит денежные средства на расчетный счет, указанный контрагентом. Копия принятого к оплате платежного поручения передается бухгалтером по возвращению в офис секретарю, который отправляет ее по факсу контрагенту с целью уведомить об оплате выставленного счета.

После уведомления контрагента об оплате или после получения им денег, последний передает (пересылает) демонстрационные материалы фирме. После их получения начинается процесс первичного освоения ИТ ответственным лицом (смотрите рис.8.6.13).

Первичное знакомство с ИТ на уровне демонстрационных материалов дает возможность специалисту понять основную концепцию данной ИТ и более точно определить область ее применения. На этом этапе определяются основные возможности ИТ, проводятся необходимые эксперименты и результаты исследований сравниваются с возможностями других ИТ.

Процесс выбора ИТ для освоения изображен на рис.8.6.14. После первичного ознакомления с ИТ его результаты обсуждаются на специальном совещании, где принимается решение о продолжении или прекращении освоения ИТ. Если возможности рассматриваемой ИТ достаточно высоки, и она по всем прогнозам является перспективной или активно развивающейся на рынке ИТ, то принимается решение отработать данную ИТ на опытной площадке.

Опытной площадкой для отработки новых ИТ является коммерческое направление деятельности фирмы в части производства посреднических услуг. Этот процесс подразделяется на подпроцесс “Внедрение ИТ в БП фирмы” и “Осуществление коммерческой деятельности”, которые протекают параллельно друг другу (на рис.8.6.12 между этими БП отсутствует связь типа “предшествует”).

Здесь мы рассмотрим подробнее только первый подпроцесс, т. е. внедрение ИТ в БП самой фирмы.

Первый этап внедрения ИТ в деятельность ООО “ААА” сводится к отработке ИТ на уровне демо-материалов. Здесь уточняется область применения ИТ в работе коммерческого направления фирмы и определяются

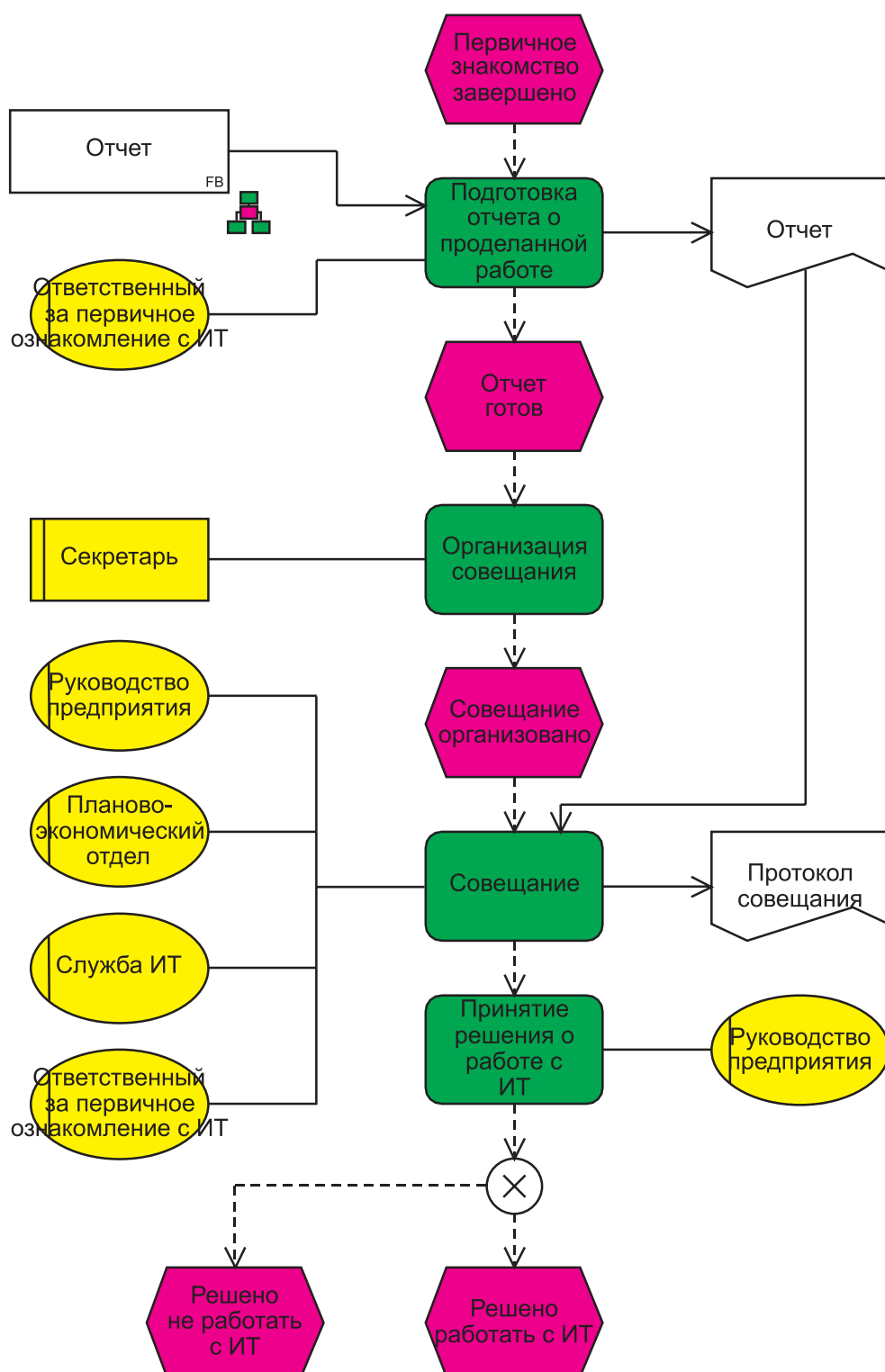


Рис.8.6.14 Выбор ИТ для освоения

сотрудники, которые будут работать в рамках данной ИТ. В процессе отработки ИТ используются демонстрационные материалы в той степени применения, в которой позволяют их ограничения. По ходу работы с ИТ ответственное лицо формирует инструкции по применению ИТ в БП фирмы и оценивает положительные и отрицательные стороны такого применения.

В случае эффективного внедрения в БП фирмы данной ИТ (на уровне демонстрационных материалов) встает вопрос о приобретении полнофункционального лицензионного варианта ИТ и сотрудничестве с авторами и распространителями данной ИТ в качестве партнеров.

Для принятия решения о заключении договоров о партнерстве и приобретения рабочих вариантов ИТ проводится совещание с участием ответственного за освоение ИТ, ЗДИТ, специалистов СИТ, главного бухгалтера, экономиста и сотрудника, который работал с данной ИТ. На этом совещании обсуждаются плюсы и минусы рассматриваемой ИТ, а также возможность и перспективность ее применения в БП финансово-хозяйственной деятельности фирмы и внедрения в фирмах-клиентах.

Если в результате проводимого совещания принимается решение о сотрудничестве с фирмой-распространителем ИТ, то начинается процесс заключения партнерского договора, который делится на подпроцесс “Предложение сотрудничества” и “Заключение договора”. Первый из них протекает по следующему сценарию.

ЗДИТ или ответственный специалист (менеджер) СИТ связывается с контрагентом с целью предложить сотрудничество в сфере продвижения ИТ на рынке. Это может осуществляться посредством самостоятельного телефонного звонка или звонка через секретаря, посредством электронной почты, посредством коммерческого предложения.

В случае получения положительного ответа (от контрагента) руководством фирмы совместно с заинтересованными специалистами анализируются условия взаимоотношений с контрагентом и принимается или не принимается решение о сотрудничестве. Если принимается решение о сотрудничестве, то руководство фирмы поручает ЗДИТ заключить договор. ЗДИТ дает соответствующую команду специалисту СИТ.

Процесс заключения договора зависит от того, кто готовит текст проекта договора. Он может готовиться фирмой и отправляться на согласование и подпись контрагенту или наоборот. В зависимости от этого меняется ход процесса и время его выполнения.

Рассмотрим процесс заключения договора, который готовит контрагент. В этом случае распространители ИТ имеют стандартные бланки подобных договоров, которые они присылают зачастую вместе с рекламны-

ми или демонстрационными материалами.

Проект договора присылает контрагент. Его принимает секретарь и распределяет по соответствующему маршруту. Происходит согласование проекта договора. В результате высказываются замечания и пожелания по содержанию текста. После ознакомления с присланным проектом договора ответственный специалист СИТ может либо составить протокол разногласий к договору, либо внести изменения в сам проект договора. После внесения изменений откорректированный проект договора передается заинтересованным лицам либо лично, либо посредством системы документооборота по маршруту согласования. Согласованный документ передается на утверждение директору и затем секретарю, который проставляет печать на документе, и после команды заместитель директора отправляет его контрагенту по факсу. Далее бумажный вариант документа размещается секретарем в папке предварительных договоров и у ответственного за этот договор запрашивается электронный вариант документа. Ответственный исполнитель пересылает документ секретарю, который размещает его в “Электронном Хранилище” в предварительных договорах.

Ожидается ответ контрагента по поводу высказанных замечаний и предложений (если они были сделаны). Контрагент сообщает о своем решении посредством телефонного звонка, который принимает секретарь и соединяет абонента с запрашиваемым лицом (сотрудником).

Таким образом, если решение контрагента положительно, то договор считается заключенным, и проводятся мероприятия по обмену оригиналами документов, а также осуществляется перевод договора из предварительных в исполняемые как в “Бумажном”, так и в “Электронном” хранилищах.

Ожидаются оригиналы от контрагента. Эти документы могут высылаться им либо почтой, либо поездом, либо с нарочным. Если оригиналы документов посланы почтой, то после их получения секретарь сообщает об этом заинтересованным лицам и передает документы по маршруту. После ознакомления с полученными документами они передаются секретарю, который размещает их в “Бумажном Хранилище”.

После заключения партнерского договора контрагентами проводится ряд контактов, в результате которых решаются вопросы предоставления лицензионных ИТ и оплаты за них. В результате достигнутых договоренностей осуществляются необходимые действия для получения лицензионных ИТ. После получения ИТ производится ее внедрение и использование в БП коммерческой и научно-производственной деятельнос-

ти фирмы с целью их отработки и освоения.

Внедрение и использование ИТ в работе фирмы осуществляется на базе отработанных демо-материалов с соблюдением необходимых правил и с вовлечением в процесс новых сотрудников, составлением инструкций по работе с ИТ. Одновременно начинается работа с клиентами по продаже и внедрению у них ИТ.

Один из описанных здесь процессов представлен в виде eEPC-диаграммы на рис.8.6.15. Диаграммы других процессов рекомендуем читателям составить самостоятельно.

Проанализируем процесс приема и первичной обработки коммерческого предложения (КП), диаграмма которого изображена на рис.8.6.16. Здесь в качестве контрагентов фирмы могут выступать как покупатели, так и поставщики. КП покупателей заключается в предложении покупки определенной партии конкретного товара по определенной цене, в необходимом количестве, требуемого качества и в оговоренные сроки. КП поставщиков заключается в их готовности поставить необходимые материалы для производства конкретной продукции согласно условиям, удовлетворяющим как предприятие, так и самих поставщиков. Отвечающие за процесс приема и обработки КП от контрагента находятся в состоянии ожидания КП.

КП может поступить в виде телефонного звонка, в виде сообщения по факсу, либо в виде электронного сообщения по почте. Телефонные звонки и сообщения по факсу принимаются секретарем предприятия, который отвечает также за составление электронных вариантов полученных КП. В первом случае секретарь соединяет торгового агента контрагента с требуемым лицом. Если по каким либо причинам требуемого лица не оказалось на рабочем месте, то секретарь фиксирует полученную информацию для последующей передачи ее требуемому лицу. После удовлетворительных переговоров контрагента с представителем предприятия и получения распоряжения о составлении КП секретарь составляет КП в электронном варианте.

После того, как электронный вариант КП составлен, секретарь размещает его в электронном хранилище, делает твердую копию КП, проводит процедуру регистрации КП. Регистрация подразумевает присвоение документу определенного входящего номера и записи в журнале входящих документов (бумажном и электронном), а также прикрепление к бумажному варианту маршрутной карточки.

После регистрации КП проводится его предварительное рассмотрение, которое предполагает фиксацию статуса КП (КП от покупателя или поставщика) и определение маршрута его дальнейшего движения. КП может

быть передано непосредственно через личный контакт, через систему документооборота либо в определенное время направлено в папку "Входящие". Если по каким-либо причинам КП не было отправлено, то процедура рассмотрения проводится повторно, т. е. процесс повторяется до тех пор, пока документ не дойдет до адресата.

Одно из главных условий повышения эффективности работы фирмы – снижение затрат на осуществление своей деятельности. К факторам, обеспечивающим снижение себестоимости, относятся экономия материальных ресурсов, повышение производительности труда, упорядочение затрат на аппарат управления и т. д., а также применение новейших технологий, в том числе информационных.

Для успешного выполнения БП приема и первичной обработки КП от контрагента фирма вынуждена совершать различные затраты. При моделировании этого БП ставилась задача комплексно взглянуть на реализацию функций в цепи БП и получить четкое и ясное представление о том, каким образом протекает БП, кто из служащих ответственен за выполнение той или иной функции БП, какие ресурсы необходимы для успешного и эффективного выполнения функции и какие затраты связаны с их выполнением.

Существует множество различных классификаций, группирующих затраты по элементам. В ARIS с Easy-фильтром затраты выражаются в денежном эквиваленте и делятся следующим образом: материальные затраты, затраты на персонал, прочие затраты. Общие затраты определяются путем суммирования вышеперечисленных затрат. Информация о затратах, связанных с выполнением конкретной функции в цепи БП вносится в атрибуты объекта и в дальнейшем является внутренней информацией этого объекта. ARIS позволяет учитывать затраты в любых (из списка доступных) величинах измерения.

При моделировании промышленных БП, т. е. БП, связанных с производством конкретного изделия, подсчет произведенных затрат не составляет особого труда (смотрите следующую главу). Затраты в стоимостном выражении получаются путем суммирования стоимости ресурсов, сырья, материалов, топлива, энергии и т. д., израсходованных на производство изделия. Известны методики расчета заработной платы производственного персонала. Затраты времени связаны, как правило, с длительностью производственного цикла, временем транспортировки и т. д.

Сложнее подсчитать затраты в офисных БП, например, в тех, которые описаны в этом разделе. В Приложениях 8, 9 представлены отчеты, созданные для этих моделей на основе сценария ProcessesAsStructogram. С помощью этих отчетов можно проводить анализ БП приема и первичной обработки КП. Используя информацию этих отчетов,

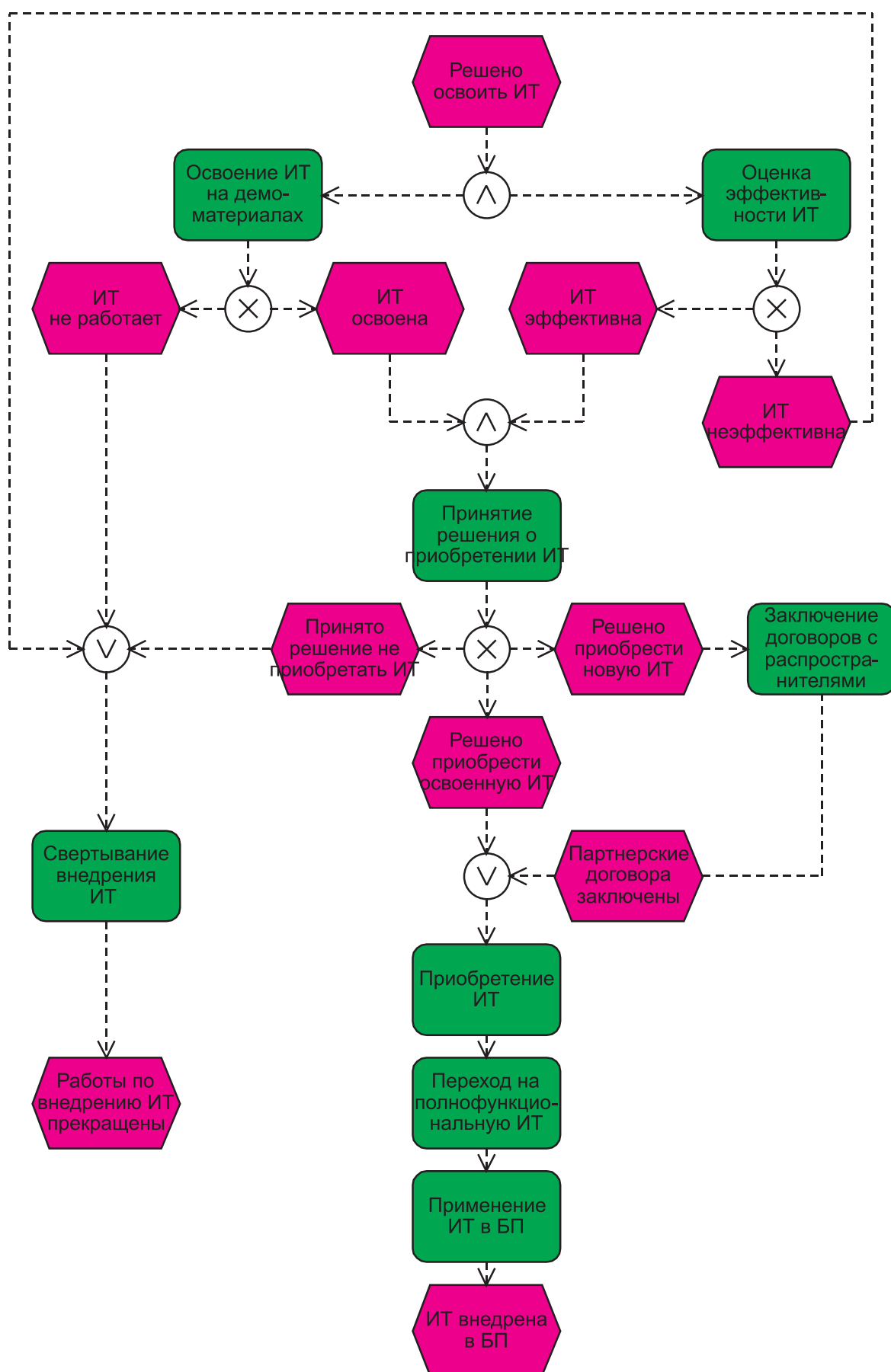


Рис.8.6.15 Внедрение ИТ в БП

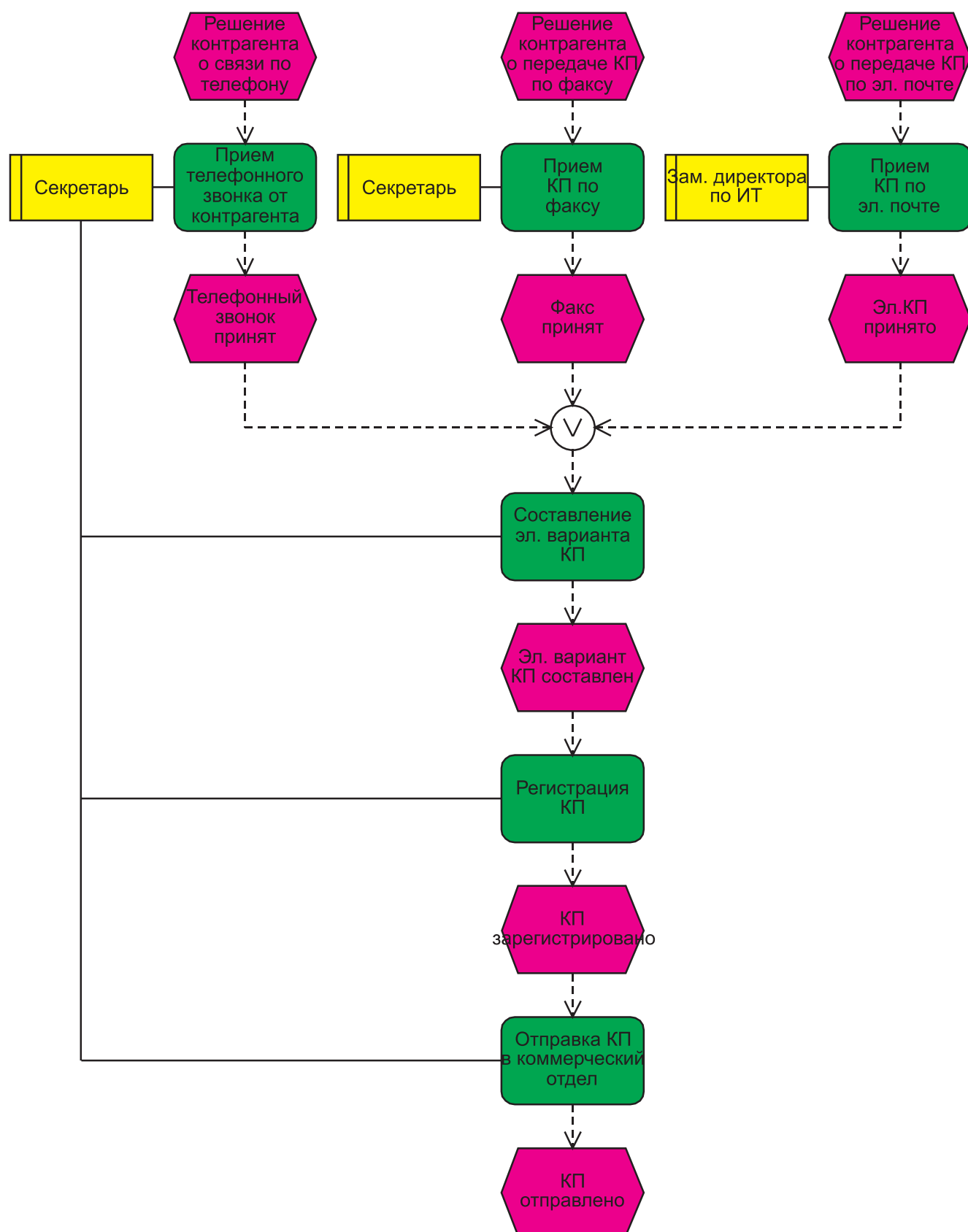


Рис.8.6.16 Прием и первичная обработка КП

можно, например, выяснить, что стоимостные затраты распределились на материальные, затраты на персонал, на

прочие и общие, которые составили 6450 руб. Все это отражено в таблице 8.6.1.

Табл. 8.6.1

| Функция                                 | Затраты (средние)   | Значение       |             |
|---|---------------------|----------------|-------------|
|   |                     | Руб.           | %           |
| Прием телефонного звонка от контрагента | Материальные        | 200,00         | 10,0        |
|   | Затраты на персонал | 415,00         | 10,0        |
|   | Прочие              | 30,00          | 10,0        |
|   | <b>Общие</b>        | <b>645,00</b>  | <b>10,0</b> |
| Прием КП по факсу                       | Материальные        | 400,00         | 20,0        |
|   | Затраты на персонал | 539,50         | 13,0        |
|   | Прочие              | 75,00          | 25,0        |
|   | <b>Общие</b>        | <b>1014,50</b> | <b>15,7</b> |
| Прием КП по эл. почте                   | Материальные        | 400,00         | 20,0        |
|   | Затраты на персонал | 1120,50        | 27,0        |
|   | Прочие              | 75,00          | 25,0        |
|   | <b>Общие</b>        | <b>1595,50</b> | <b>24,7</b> |
| Составление эл. варианта КП             | Материальные        | 200,00         | 10,0        |
|   | Затраты на персонал | 622,50         | 15,0        |
|   | Прочие              | 30,00          | 10,0        |
|   | <b>Общие</b>        | <b>852,50</b>  | <b>13,2</b> |
| Регистрация КП                          | Материальные        | 600,00         | 30,0        |
|   | Затраты на персонал | 1037,50        | 25,0        |
|   | Прочие              | 60,00          | 20,0        |
|   | <b>Общие</b>        | <b>1697,50</b> | <b>26,3</b> |
| Отправка КП в коммерческий отдел        | Материальные        | 200,00         | 10,0        |
|   | Затраты на персонал | 415,00         | 10,0        |
|   | Прочие              | 30,00          | 10,0        |
|   | <b>Общие</b>        | <b>645,00</b>  | <b>10,0</b> |
| <b>Итого</b>                            |                     | <b>6450,00</b> |             |

Анализ БП приема и первичной обработки КП от контрагента показал, что наибольшую долю в составе общих затрат составляют материальные затраты и затраты на персонал. Поэтому необходимо стремиться к снижению стоимости этих составляющих БП при сохранении необходимого качества.

Оптимизация БП может быть произведена за счет сокращения объема документации и отчетности, ускорения документооборота и потока данных, внедрения ИТ для улучшения процесса обработки информации. Для этого необходимо на всех этапах БП разработать типовые формы и правила оформления документации. Некоторые функции, выполняемые служащими фирмы, может выполнять ПО. Например, выбор маршрута и передача электронных документов адресату может производиться не секретарем фирмы, а ПО. Таким образом, секретарю необходимо только составить электронные варианты КП контрагентов и поместить их в

хранилище данных. ПО само отправит информацию по нужному маршруту, что приводит к сокращению времени на выполнение функций в цепи БП.

В заключение проанализируем БП «Обработка договора по коммерции». Модель этого БП представлена на рис.8.6.17. Из нее видно, что договор по коммерции проходит достаточно длинный маршрут согласования и утверждения. Недостаток данного БП связан с большими временными и материальными затратами. Например, после подготовки проекта договора менеджер договора (МД) передает его для согласования экономисту. Если у последнего есть замечания, то он их указывает и возвращает документ на доработку. МД, получив документ и замечания, редактирует его, распечатывает и снова передает экономисту. После чего экономист осуществляет еще одну проверку и, если нет замечаний, передает документ на согласование главному бухгалтеру, который поступает точно так же как экономист. То



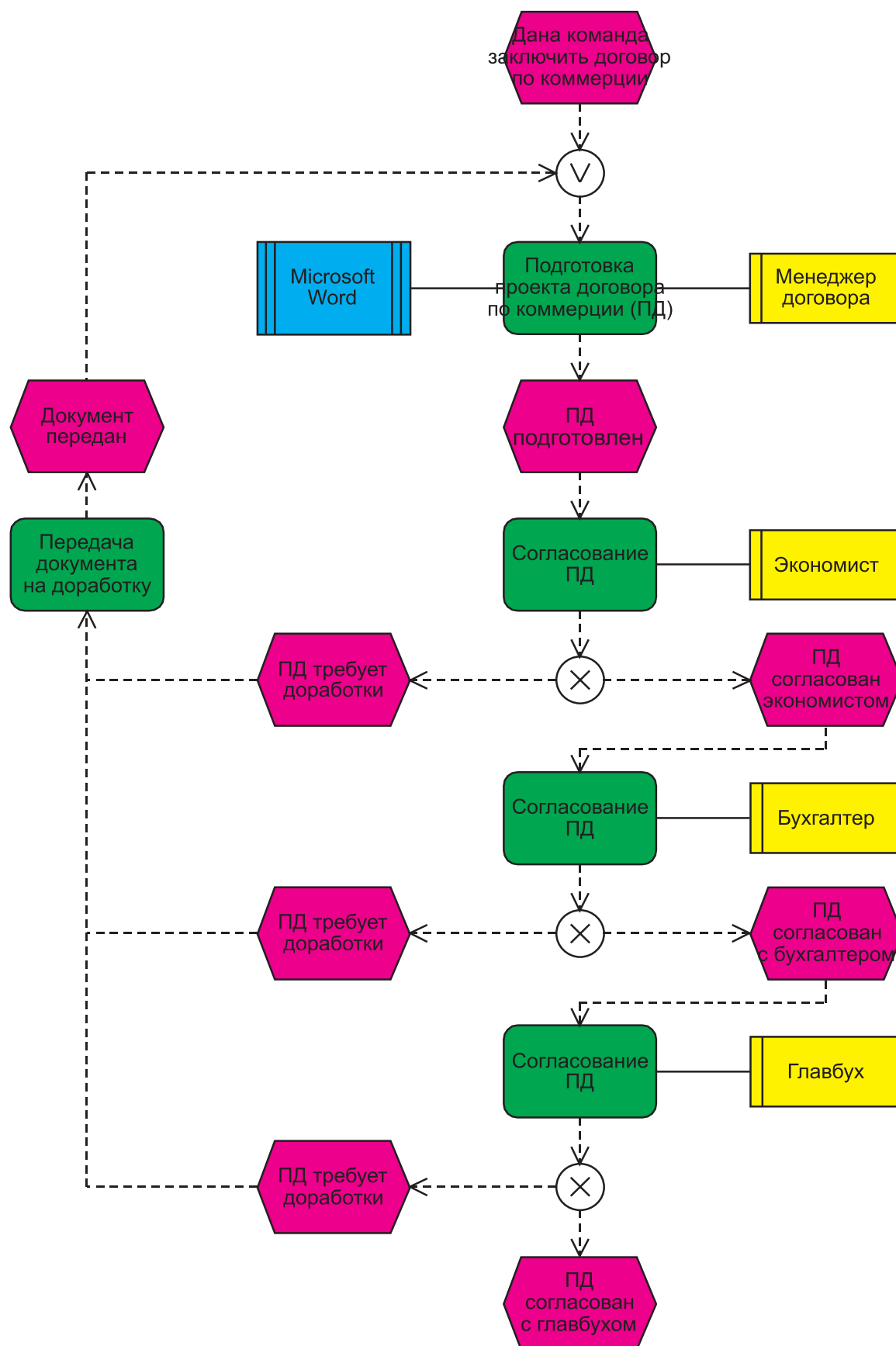


Рис.8.6.17 Обработка договора по коммерции (начало)

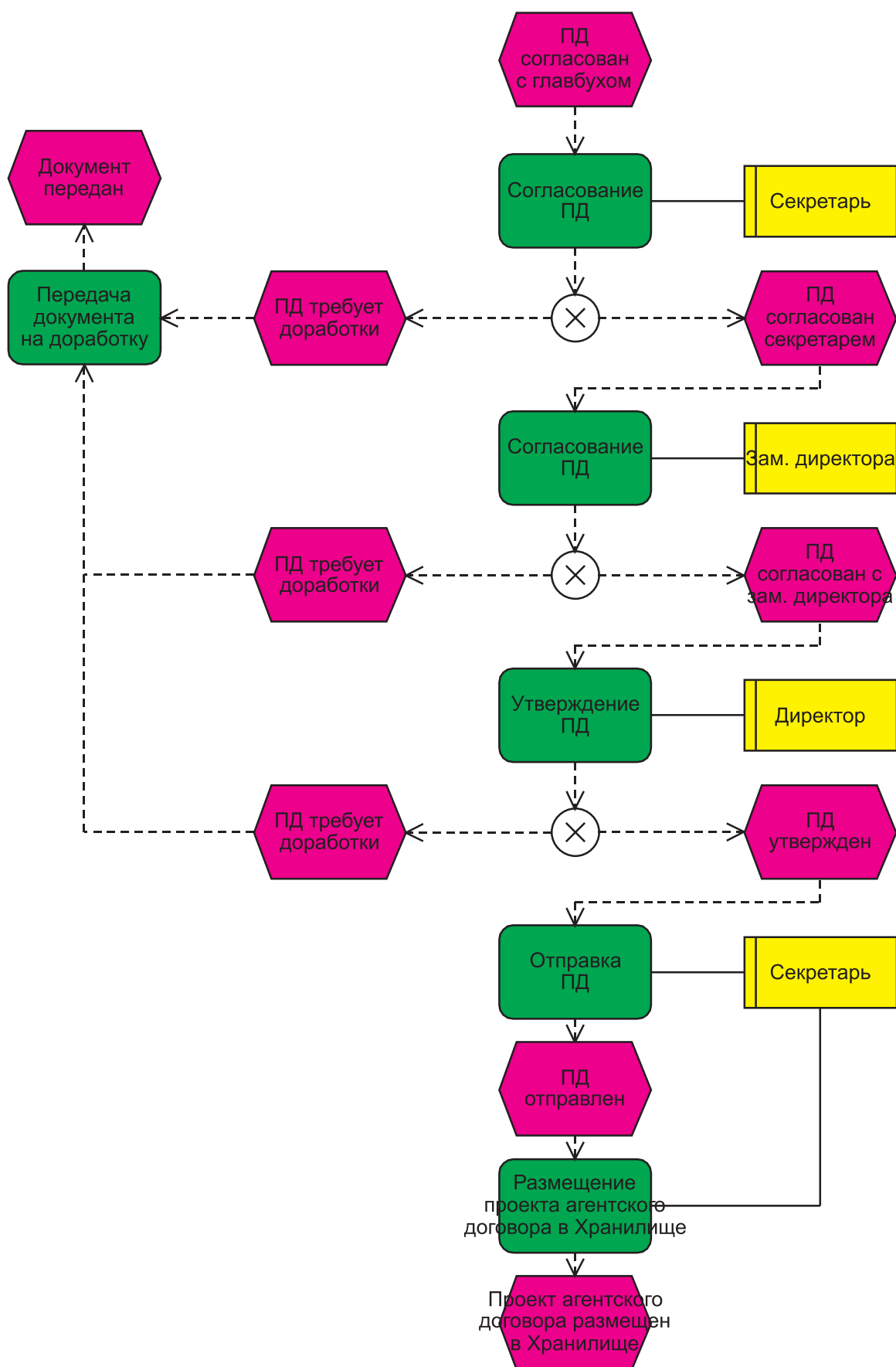


Рис.8.6.17 Обработка договора по коммерции (окончание)

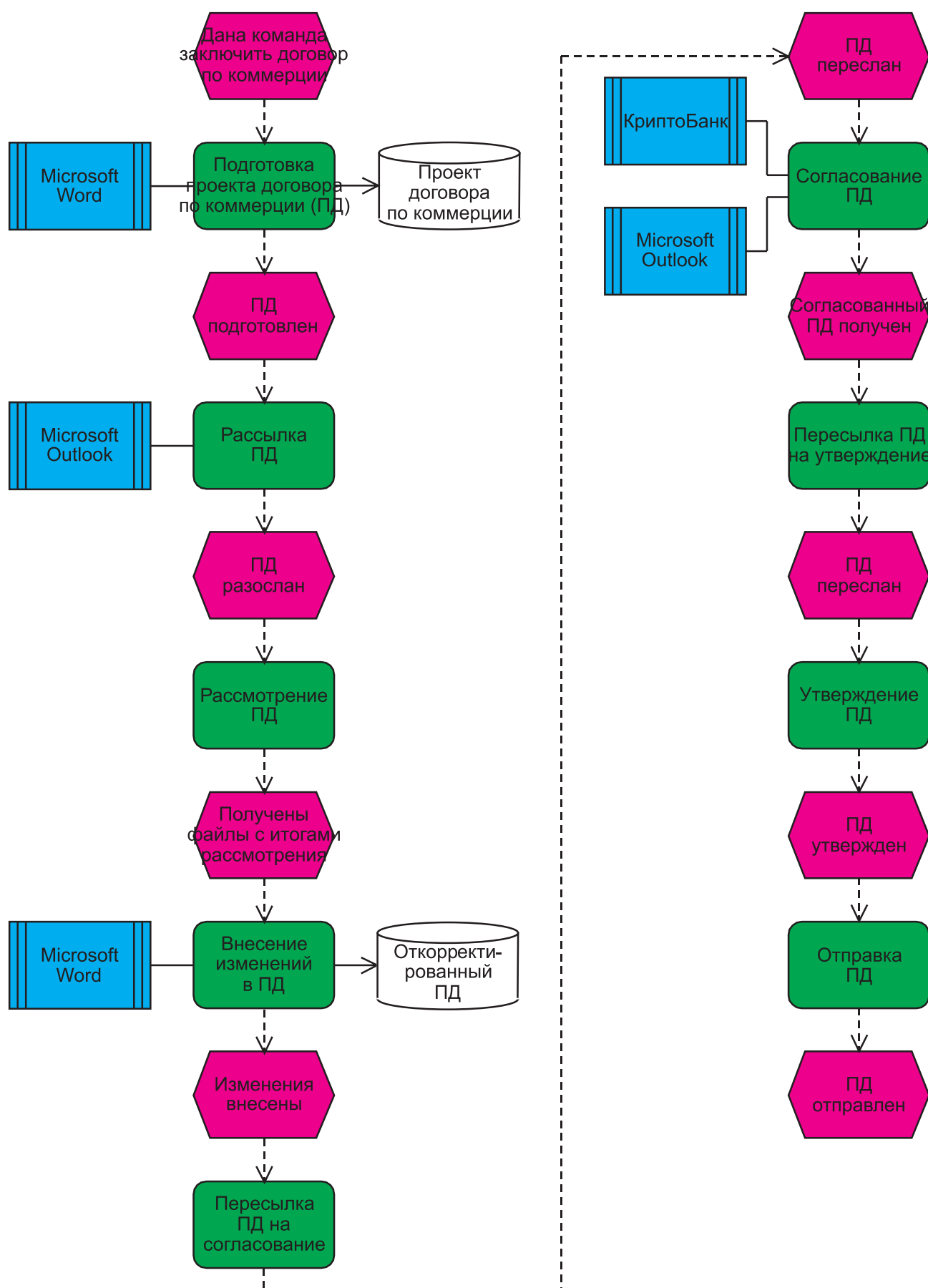


Рис.8.6.18 Улучшенная модель "Обработка договора по коммерции"

есть, если у него есть замечания, то документ возвращается МД, он вносит исправления, распечатывает его и снова отдает на согласование по маршруту. Таким образом, по всему маршруту согласования имеют место излишние затраты времени и средств в виде бумаги и расходных материалов печатающего устройства.

В результате анализа этого БП можно сделать выводы, что работа субъектов ответственности по выполнению функции БП связана, в основном, с общей обработкой документов и их оформлением. Повышение эффективности этого БП связано с обеспечением сквозного доступа к документам без их дублирования, персональной обработкой полученных данных, совмещением электронной и вербальной коммуникации и т. д. При выполнении своих обязанностей субъекты ответственности должны иметь возможность дистанционной и совместной работы над документами в локальной сети и поддерживать способы общения, не покидая рабочего места. Поэтому при оптимизации этого БП необходимо сосредоточить внимание на более эффективном управлении архивом документов, управлении системой маршрутизации пакетов документов, обеспечение совместной работы над электронными документами и т. д.

В результате усовершенствования предложена модель рассматриваемого БП, которая изображена на

рис.8.6.18. Из нее видно, что после подготовки проекта договора по коммерции МД рассылает файл с этим документом по электронной почте всем участникам маршрута согласования для рассмотрения. Последние, получая данный файл, осуществляют знакомство с ним, указывают в электронном виде свои замечания (если они есть) и отправляют документ обратно МД. Он корректирует исходный вариант текста проекта договора с учетом всех присланных поправок. Затем МД пересылает документ по маршруту согласования, и участники этого процесса осуществляют согласование и визирование документа посредством электронной подписи с использованием соответствующего ПО. Завизированный таким образом документ возвращается к МД, и он отправляет его на утверждение директору. Последний также осуществляет визирование документа электронной подписью и пересылает его секретарю, который распечатывает документ. Затем секретарь передает бумажный вариант документа директору, последний визирует его и возвращает секретарю для проставления печати, отправки документа и размещения его в Хранилище.

Рассмотренные примеры показывают, что в результате составления моделей БП и их анализа можно значительно повысить их эффективность.

## 9. МОДЕЛИРОВАНИЕ ПРОИЗВОДСТВЕННО-СБЫТОВОЙ ФИРМЫ

Экономико-математические методы становятся важным инструментом разработки оптимальных планов развития предприятия, а также качественного их выполнения. Они используются там, где нужны детальные математико-статистические прогнозы спроса на продукцию и сырье, оптимизация размещения капиталовложений, точный учет экологических последствий производственной деятельности, анализ движения трудовых ресурсов и т. д. Сегодня экономика широко использует математику и становится точной наукой. В этой связи уместно вспомнить высказывание К. Маркса: «Наука только тогда достигает совершенства, когда ей удается пользоваться математикой».

Экономико-математические методы – это новая технология управления производством. Поэтому экономисты часто относятся к ним как к любой новой технике и технологии, т. е. отторгают их. Это происходит в том числе по вине самих экономико-математических методов, так как многим из них не хватает простоты и наглядности изложения, нет кратких практических методик по их применению.

Многие экономические объекты могут быть достаточно точно представлены в виде экономико-математической модели. Эти модели позволяют планировать деятельность предприятия и путем управления обеспечивать выполнение этих планов. Задачи планирования сводятся к определению объема выпуска продукции, к выбору номенклатуры изделий, к согласованию плановых показателей с ресурсами и т. д. При управлении предприятием решаются вопросы оптимизации затрат, назначения цен, распределения прибыли, определения источников финансирования, ускорения сбыта продукции и т. д.

В этой главе рассматривается ARIS-моделирование финансово-обеспеченной деятельности производственно-сбытовой фирмы, где все расчеты основаны на выведенном балансовом уравнении и формулах вычисления выручки, прибыли, постоянных и переменных затрат, а также вычисления суммы кредита и ежепериодных платежей за их погашение.

### 9.1. Основное балансовое уравнение

Рассмотрим расчетные формулы для моделирования производственно-сбытовой фирмы и процессов финансово-обеспеченного производства, т. е. производства, в котором затраты сбалансированы собственными финансовыми средствами и всеми видами поступлений. Для учета всех внутренних и внешних факторов, связанных с обеспечением (снабжением) производством и реализацией продукции, заложим в модель возможность ее дальнейшего функционального наполнения. Такое наполнение необходимо для учета простого и расширенного воспроизводства, конъюнктуры рынка, изменения законодательства, форсмажорных обстоятельств, изменения ставки рефинансирования центрального банка РФ и т. д. Математическая модель должна быть гибкой, настраиваемой и адаптируемой под конкретные внутренние и внешние производственно-сбытовые условия. Для этого необходимо выбрать основное (базовое) уравнение, которое включает величины, описывающие основные аспекты производственной, сбытовой, финансовой и другой деятельности предприятия.

Чтобы не усложнять задачу, построим математическую модель финансово-обеспеченного производства для предприятия, которое производит и реализует один вид продукции, имея для этого достаточные производственные мощности.

При финансово-обеспеченном производстве должно выполняться следующее соотношение: *Затраты* = *Поступления*. Конкретизируем это соотношение.

Затраты на производство продукции делятся на переменные и постоянные. *Переменные затраты* зависят от объема производимой продукции и включают затраты на сырье, материалы, комплектующие изделия, топливо, зарплату производственного персонала с учетом налогов и обязательных платежей. *Постоянные затраты* не зависят от объема производимой продукции и включают арендную плату (или налоги) за землю, помещения и оборудование, платежи в погашение ранее полученных кредитов, плату за телефон, затраты на охрану, зарплату непроизводственному персоналу с учетом налогов и платежей на нее, затраты службы маркетинга, сбыта и т. д.

Основными источниками получения финансовых средств является выручка от произведенной и реализо-

ванной продукции, а также собственные (резервные) средства предприятия и заемные средства, например, кредиты банков. Таким образом, имеет место следующее более детальное базовое уравнение:

$$\begin{aligned} \text{переменные затраты} + \text{постоянные затраты} = \\ = \text{собственные средства} + \text{заемные средства} \\ + \text{выручка}. \end{aligned}$$

Базовое уравнение в левой части содержит затраты, а в правой – поступления, их компенсирующие. Затраты разделены на две части: переменные (связанные с объемом производства) и постоянные. В общем случае левую часть можно записать в следующем математическом виде  $U(X) + Z$ , где  $U(X)$  – функция переменных затрат, зависящая от объемов производства  $X$ ,  $Z$  – величина постоянных затрат. В частном, но наиболее распространенном случае функция  $U(X)$  является линейной и представляется в виде  $U * X$ , где  $U$  – коэффициент пропорциональности, не зависящий напрямую от  $X$ .

В правой части базового уравнения содержатся слагаемые, которые описывают средства, привлекаемые для производства продукции в расчетный период. Знак равенства между левой и правой частью указывает на то, что поступления компенсируют затраты, т. е. мы моделируем финансово-обеспеченное производство. Если обе части этого уравнения измерять в денежном выражении, то с учетом его структуры можно сказать, что это уравнение является балансовым. Первым слагаемым в правой части являются собственные (резервные) средства существующего предприятия или средства организаторов и акционеров нового (организуемого) предприятия.

Балансовое уравнение (БУ) явно содержит объемы производства и поэтому, в основном, используется для их вычисления. В случае линейной зависимости переменных затрат от объемов производства искомая величина  $X$  вычисляется по формуле  $X = (T - Z) / U$ , где  $T$  – сумма всех средств (собственных, заемных и выручки), которые привлекаются для производства продукции в объеме  $X$ ,  $U$  – переменные затраты на одно изделие.

Величина переменных затрат  $U$  описывает все категории затрат, связанных с объемами производства, в частности, стоимость комплектующих изделий и материалов, зарплату производственного персонала и многое другое. Через эту величину можно функционально наполнять базовую модель при учете взаимодействия с поставщиками, при моделировании незавершенного производства и т. д.

Величина постоянных затрат БУ позволяет моде-

лировать (оценивать) основные фонды предприятия, описывать затраты на их содержание и развитие, в том числе моделировать работу маркетинговой службы, аппарата управления, охраны, обслуживание складских помещений и т. д.

Записанное БУ статично. Для придания ему динамических черт введем в обращение период  $i$  (месяц, квартал, год или любой другой отрезок времени) и следующие обозначения:  $C_i^r$  – собственные (резервные) финансовые средства предприятия, которые используются для производства продукции в  $i$ -й период;  $C_i^d$  – заемные средства, которые используются для производства продукции в  $i$ -й период;  $S_i$  – выручка в  $i$ -й период;  $X_i$  – объем продукции (например, количество в штуках), произведенной предприятием в  $i$ -й период;  $Z_i$  – постоянные затраты производства в  $i$ -й период;  $U_i$  – переменные затраты, приходящиеся на одно изделие в  $i$ -й период. Выбирая символы введенных обозначений мы ориентировались на первые буквы следующих слов: reserve (резерв), debt (заем), sales (продажа, выручка), unit costs (единичные затраты). В этих обозначениях БУ примет вид

$$U_i X_i + Z_i = C_i^r + C_i^d + S_i. \quad (1)$$

В левой части этого уравнения баланса финансовых средств находятся требуемые затраты, а в правой – используемые в  $i$ -й период резервные средства и поступившие в  $i$ -й период финансовые средства.

## 9.2. Вычисление выручки и каноническая форма балансового уравнения

Выручку в  $i$ -й период составляют суммы денег от продажи продукции в этот период. В  $i$ -й период могут быть проданы продукты, произведенные как в  $i$ -й, так и в предшествующие периоды. Цена продукции в общем случае зависит от периода ее изготовления и от текущего ( $i$ -го) периода. Поэтому в общем случае

$$S_i = a_i C_i X_i + \sum_{k=1}^{N_i} C_{ik} X_{ik}, \quad (2)$$

где  $N_i$  – число периодов хранения ранее выпущенных, но реализуемых в  $i$ -й период продуктов;  $C_{ik}$  – цена продукта, изготовленного в  $(i - k)$ -й период, но проданного в  $i$ -й период;  $X_{ik}$  – объем продукции, выпущенной в  $(i - k)$ -й период, но проданной в  $i$ -й период;  $C_i$  – цена продукта в  $i$ -й период;  $X_i = X_{i0}$ ;  $a_i$  – коэффициент реализации в  $i$ -й период продукции,

изготовленной в этот же период  $0 \leq a_i \leq 1$ .

Величину  $C_{ik}$  можно выразить через цену продукции в  $i$ -й период, т. е. записать ее в виде  $C_{ik} = b_{ik}C_i$ , где  $b_{ik}$  – коэффициент влияния на цену в  $i$ -й период времени хранения ( $k$  периодов) продукта. Например, для хороших сортов вина  $b_{ik} \geq 1$  и  $b_{ik}$  растет с увеличением  $k$ . Для портящихся продуктов  $0 \leq b_{ik} \leq 1$ .

Величину  $X_{ik}$  можно записать в виде  $X_{ik} = a_{ik}X_{i-k}$ , где  $a_{ik}$  – коэффициент реализации (продажи) продуктов, изготовленных в  $(i-k)$ -й период, выручка от которых поступает на счет предприятия в  $i$ -й период;  $X_{i-k}$  – объем (количество) продуктов, выпущенных за  $k$  периодов до расчетного. С использованием коэффициентов  $a_{ik}$ ,  $b_{ik}$  выражение (2) можно записать в виде

$$S_i = a_i C_i X_i + C_i \sum_{k=1}^{N_i} b_{ik} a_{ik} X_{i-k}. \quad (3)$$

Если число периодов хранения (реализации) постоянно, а цена продукции в расчетный период не зависит от периода ее изготовления ( $b_{ik} = 1$ ), то

$$S_i = C_i \sum_{k=0}^N a_{ik} X_{i-k}, \quad (4)$$

где  $N$  – число периодов реализации продукции, выпущенной в любой из периодов;  $a_{i0} = a_i$  – коэффициент реализации продукции, изготовленной в  $i$ -й период. В рассматриваемом случае

$a_{i0} + a_{i1} + a_{i2} + \dots + a_{iN} = 1$ ,  
т. е. за  $N$  периодов реализуется вся продукция.  
С учетом (4) БУ (1) записывается в виде

$$U_i X_i + Z_i = C_i^r + C_i^d + C_i \sum_{k=0}^N a_{ik} X_{i-k}. \quad (5)$$

Это уравнение характеризует возможности предприятия приобрести все необходимые (для производства продукции в объеме  $X_i$ ) материальные ресурсы, оплатить труд персонала предприятия, сделать отчисления на компенсацию израсходованных основных фондов производства (на амортизацию оборудования и зданий), оплатить постоянные расходы предприятия.

Величину  $X_{i-k}$  можно выразить через достигнутый уровень производства, т. е. представить в виде  $X_{i-k} = r_{ik} X_{i-1}$ , где  $r_{ik}$  – коэффициент влияния времени ( $k$  периодов) на объемы производства, описывающий динамику производства (например, если  $r_{ik} = 1.1$ , то объемы производства от периода к периоду увеличиваются на 10%). Тогда выражение (4) можно представить в виде

$$S_i = C_i (a_i X_i + r_i X_{i-1}), \quad (6)$$

где

$$r_i = \sum_{k=1}^{N_i} b_{ik} a_{ik} r_{ik}, \quad r_{ik} = X_{i-k} / X_{i-1}.$$

С учетом этих равенств БУ (1) можно записать в виде  $(U_i - a_i C_i) X_i = C_i^r + C_i^d - Z_i + r_i C_i X_{i-1}$ , или в виде

$$X_i = A_i X_{i-1} + D_i, \quad (7)$$

где

$$A_i = r_i C_i / (U_i - a_i C_i), \\ D_i = (C_i^r + C_i^d - Z_i) / (U_i - a_i C_i).$$

Вид (7) будем называть каноническим.

На рис.9.2.1 представлена модель технических терминов для понятий БУ.

### 9.3. Вычисление объема производства и цены продукции

По определению разность выручки и затрат является *валовой прибылью* (Gross profit – валовая прибыль). Поэтому, взяв за основу, например, выражение (5), получим для прибыли предприятия в  $i$ -й период следующую расчетную формулу:

$$G_i = S_i - R_i = C_i \sum_{k=0}^N a_{ik} X_{i-k} - U_i X_i - Z_i,$$

$$\text{т. к. } R_i = U_i X_i + Z_i.$$

Если предприятие имеет нулевую прибыль, то из уравнения  $G_i = 0$  получим выражение

$$X_i = \frac{1}{C_i a_{i0} - U_i} \left( Z_i - C_i \sum_{k=1}^N a_{ik} X_{i-k} \right).$$

По этой формуле вычисляется финансово-обеспеченный объем производства продукции в  $i$ -й период, учитывающий реальные условия функционирования предприятия, работающего без прибыли. Прибыльное производство имеет место при условии  $G_i > 0$ , т. е. при

$$X_i > \frac{1}{C_i a_{i0} - U_i} \left( Z_i - C_i \sum_{k=1}^N a_{ik} X_{i-k} \right).$$

Отношение  $G_i$  к  $R_i$  называется *относительной прибылью*  $\rho_i$ :

$$\rho_i = \frac{G_i}{R_i} = \frac{S_i - R_i}{R_i} = \frac{S_i}{R_i} - 1 = \frac{C_i \sum_{k=0}^N a_{ik} X_{i-k}}{U_i X_i + Z_i} - 1. \quad (8)$$



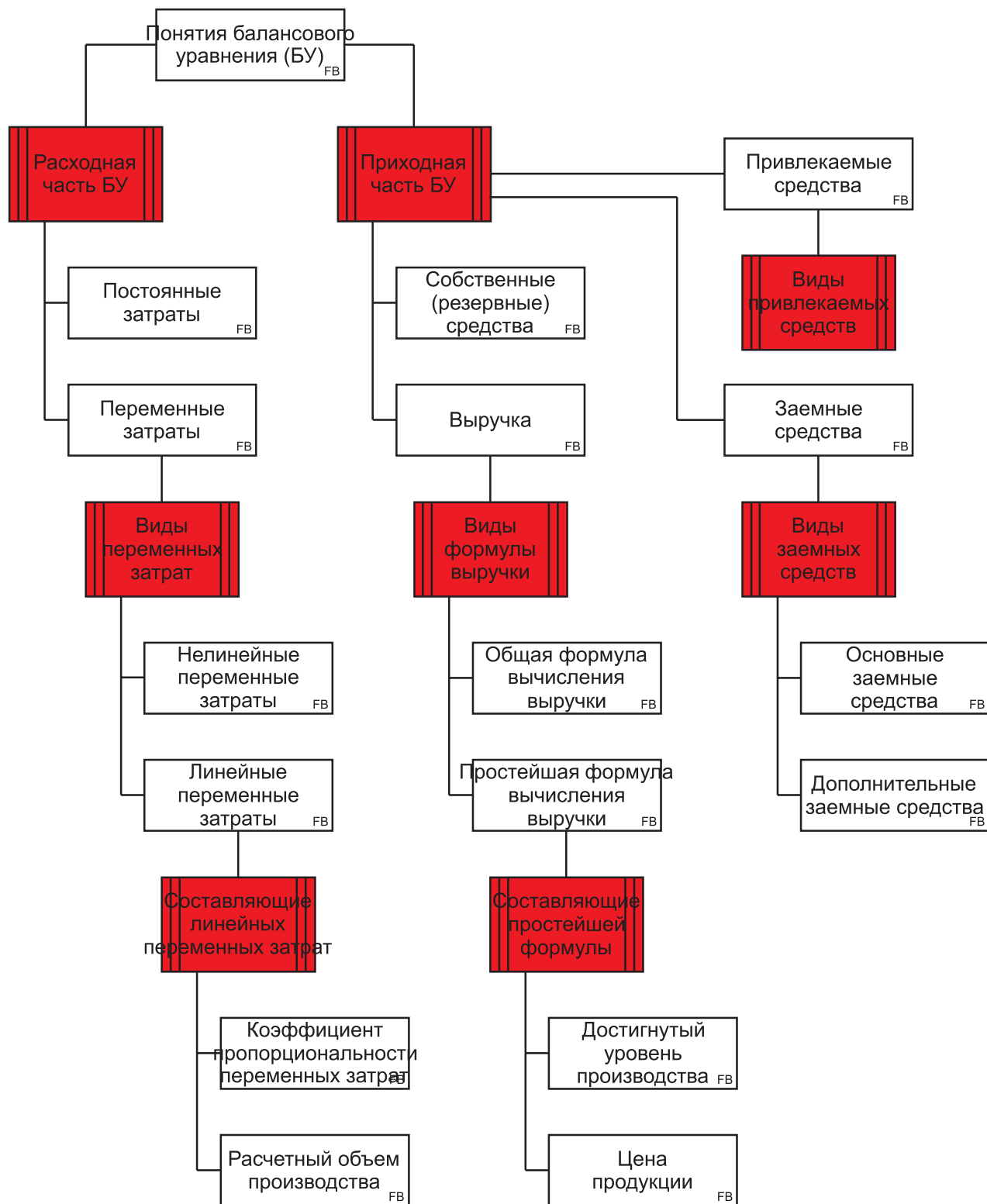


Рис.9.2.1. Модель технических терминов для понятий балансового уравнения

Если  $\rho_i > 0$ , то эту величину называют *индексом прибыли*. Если  $\rho_i < 0$ , то эту величину называют *индексом убытка*.

При нулевой прибыли в  $i$ -й период ( $\rho_i = 0$ ) из (8) получим, что цена  $C_i$  определяется по формуле:

$$C_i = \frac{U_i X_i + Z_i}{\sum_{k=0}^N a_{ik} X_{i-k}} \quad (9)$$

Если производство равномерное, т. е. объёмы производства постоянны

$$X_i = X_{i-1} = X_{i-2} = \dots = \text{const},$$

то последняя формула примет следующий вид:

$$C_i = \frac{U_i X_i + Z_i}{X_i \sum_{k=0}^N a_{ik}} = U_i + \frac{Z_i}{X_i} \quad (10)$$

Из этой формулы видно, что цена одного продукта в бесприбыльном производстве равна сумме переменных и постоянных затрат на один продукт.

$$\text{т. к. } \sum_{k=0}^N a_{ik} = 1.$$

Для прибыльного производства, т. е. когда  $\rho_i > 0$ , из (8) получим:

$$C_i = \frac{(\rho_i + 1)(U_i X_i + Z_i)}{\sum_{k=0}^N a_{ik} X_{i-k}} \quad (11)$$

т. е. цена продукции увеличивается в  $(\rho_i + 1)$  раз по сравнению с бесприбыльным производством. При равномерном производстве формула (11) примет следующий частный вид:

$$C_i = \frac{(\rho_i + 1)(U_i X_i + Z_i)}{X_i}.$$

Из полученных формул вычисления цены продукции  $C_i$  видно, что она прямо пропорциональна затратам  $U_i X_i + Z_i$ .

**Пример.** Предприятие производит продукцию машиностроения. Переменные затраты на одно изделие  $U_i = 600$  тыс. руб. Постоянные затраты  $Z_i = 20$  млн. руб. Определить, какой должна быть цена изделия  $C_i$  при равномерном выпуске продукции в случае безубыточного производства – 100 изделий, 200 изделий, 500 изделий – и производства, при котором обеспечивается индекс прибыли  $\rho_i = 0.3$ , для  $X_i = 100$ ,  $X_i = 200$ ,  $X_i = 500$ .

**Решение.** Рассмотрим случай безубыточного произ-

водства ( $\rho_i = 0$ ). Цена вычисляется по формуле (10).

Для  $X_i = 100$  получим

$$C_i = (600 * 10^3 * 100 + 20 * 10^6) / 100 = 8 * 10^5 \text{ (руб./изд.)}.$$

Для  $X_i = 200$  получим

$$C_i = (600 * 10^3 * 200 + 20 * 10^6) / 200 = 7 * 10^5 \text{ (руб./изд.)}.$$

Для  $X_i = 500$  получим

$$C_i = (600 * 10^3 * 500 + 20 * 10^6) / 500 = 64 * 10^4 \text{ (руб./изд.)}.$$

Для случая прибыльного производства  $\rho_i = 0.3$ .

Для  $X_i = 100$  получим  $C_i = (1 + 0.3) * 8 * 10^5$ .

Следовательно  $C_i = 10.4 * 10^5$  (руб./изд.).

Для  $X_i = 200$  получим

$$C_i = 1.3 * 7 * 10^5 = 9.1 * 10^5 \text{ (руб./изд.)}.$$

Для  $X_i = 500$  получим

$$C_i = 1.3 * 6.4 * 10^5 = 8.32 * 10^5 \text{ (руб./изд.)}.$$

Из анализа полученных решений видно, что цены, обеспечивающие одинаковый индекс прибыли, уменьшаются с увеличением  $X_i$ .

## 9.4. Моделирование объемов выпуска продукции

На рис.9.4.1 представлен функциональный поток процесса имитационного моделирования объемов выпускаемой продукции для финансово-обеспеченного производства. Все необходимые расчеты основаны на каноническом БУ (7), в котором неизвестной (искомой) величиной в расчетный период является объем выпускаемой продукции  $X_i$ , а все остальные параметры являются управляемыми, т. е. задаются пользователем модели или вычисляются по заданным формулам.

В качестве исходных параметров перед началом моделирования необходимо задать значения следующим величинам:

$N$  – число периодов хранения на складе ранее выпущенной продукции. Например, если  $N = 2$ , то на складе хранится продукция, выпущенная за два периода до расчетного, но еще не полностью реализованная. Причем, в  $i$ -й период будет полностью реализована продукция, выпущенная за два периода до  $i$ -го.

$X_0$  – объем достигнутого уровня производства продукции, т. е. объем продукции, произведенной в период, предшествующий первому.

$X_{-1}$  – объем продукции, выпущенной за два периода до первого.

$X_{-2}, X_{-3}, \dots$  – объемы продукции, произведенной за 3, 4, ... периодов до первого.

$a_1$  – коэффициент реализации продукции, изготов-

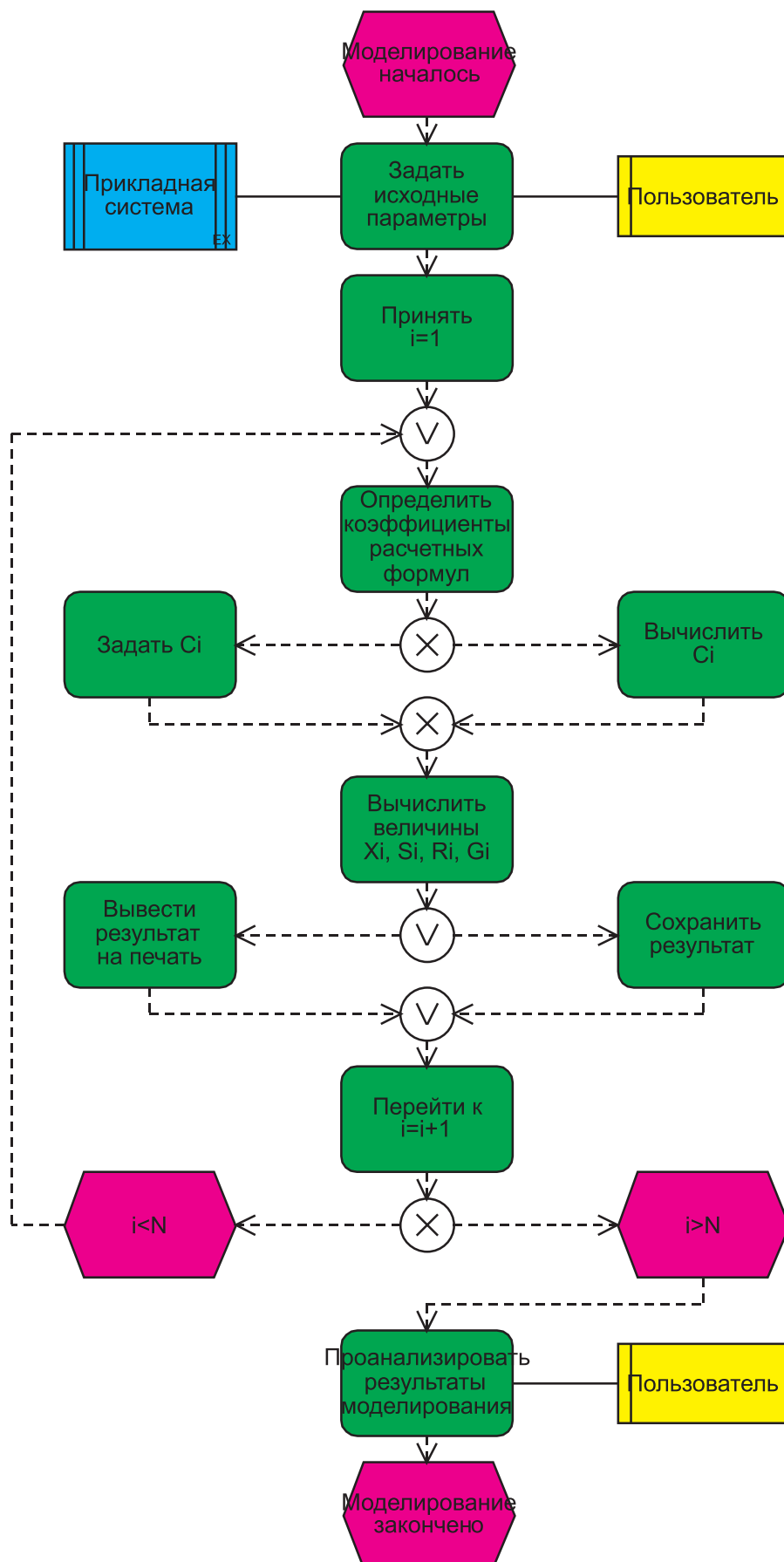


Рис. 9.4.1. Функциональный поток моделирования объемов выпуска продукции.

ленной в первый период.

$U_i, Z_i, C_i, C_i^r, C_i^d$  – управляемые (задаваемые) пользователем величины в  $i$ -й период (для  $i=1, 2, 3, \dots$ ).

Будем считать, что независимо от номера расчетного периода (летом и зимой) число периодов хранения уменьшает стоимость продукта по линейному закону, т. е.

$$b_{ik} = (N - k)/N. \quad (12)$$

Из этой расчетной формулы видно, что в период изготовления ( $k = 0, b_{ik} = 1$ ) цена максимальна и равна  $C_i$ , где  $i$  – номер периода. Но если продукция пролежит на складе периодов ( $k = N$ ), то она будет иметь нулевую цену ( $b_{ik} = 0$ ).

Основная расчетная формула в рассматриваемом случае имеет вид (7), где

$$r_i = \frac{1}{NX_{i-1}} \sum_{k=1}^{N-1} (N - k) a_{ik} X_{i-k}. \quad (13)$$

Все моделируемые величины, включая исходные, будем записывать в таблицу, в которой номер строки равен номеру моделируемого периода, а имя столбца соответствует имени задаваемой или вычисляемой переменной. Причем, сначала записываются задаваемые (управляемые) пользователем или вычисляемые по известным формулам величины, а в последних четырех столбцах записываются вычисляемые на основе БУ и по определению искомые величины  $X_i, R_i, S_i, G_i$ . По этим вычисленным значениям во время моделирования строятся графики, необходимые для анализа и принятия управленческих решений.

В качестве примера запишем алгоритм моделирования, построенный на основе описанных формул. Положим  $N = 2$ , т. е. на складе хранится часть продукции, которая выпущена за 2 периода до расчетного. В таблице 9.4.1 и на рис.9.4.2, 9.4.3 представлены исходные данные и результаты моделирования.

Перейдем к рассмотрению более содержательных примеров, в которых использование ARIS-моделей позволяет наглядно описать моделируемую экономическую систему и сформировать необходимую базу для входной, промежуточной и результатной информации.

## 9.5. Вычисление затрат

Переменные и постоянные затраты производства  $U_i$  и  $Z_i$  относятся к числу параметров, на которые руководители предприятий могут активно воздействовать путем совершенствования технологии производства (уменьшения материалоемкости, энергоемкости, механизации и автоматизации производства и т. д.), повышения производительности труда, совершенство-

вания заработной платы, изменения численности персонала. Поэтому контроль этих показателей крайне важен для повышения эффективности производства.

Конкретизируем формулы вычисления величин  $U_i$  и  $Z_i$ .

Величину  $U_i$  можно представить следующим выражением:

$$U_i = \sum_{j=1}^{N_q} C_{ij}^q q_{qj} + \sum_{j=1}^{N_w} C_{ij}^w q_{wj} + C_{ei} q_{ei} + C_{fi} q_{fi} + C_{si} \left( 1 + \sum_{j=1}^4 \beta_j \right) - C_{bi}, \quad (14)$$

где

$N_q$  – число типов комплектующих изделий, используемых для производства единицы продукции;

$C_{ij}^q$  – стоимость одного комплектующего изделия  $j$ -го типа, используемого для производства продукции в  $i$ -й период;

$q_{qj}$  – количество комплектующих изделий  $j$ -го типа в единице продукции;

$N_w$  – количество материалов, используемых для производства одного изделия;

$C_{ij}^w$  – стоимость единицы (веса, площади, объема)  $j$ -го материала в  $i$ -й период;

$q_{wj}$  – расход  $j$ -го материала на производство одного изделия;

$C_{ei}$  – стоимость единицы (кВт/ч) электроэнергии в  $i$ -й период;

$q_{ei}$  – расход электроэнергии на изготовление единицы продукции;

$C_{fi}$  – стоимость единицы веса (объема) топлива в  $i$ -й период;

$q_{fi}$  – расход топлива на изготовление единицы продукции;

$C_{si}$  – фонд заработной платы персонала (занятого в производстве) на единицу продукции в  $i$ -й период;

$\beta_1$  – индекс платежей в Пенсионный фонд;

$\beta_2$  – индекс платежей в Фонд социального страхования;

$\beta_3$  – индекс платежей в Фонд медицинского страхования;

$\beta_4$  – индекс платежей в Государственный фонд занятости;

$C_{bi}$  – стоимость возвратных средств на одно изделие от реализованных отходов в  $i$ -й период.

Для вычисления постоянных издержек производства можно использовать соотношение:

Таблица 9.4.1

| i  | $U_i$ | $Z_i$  | $C_i$ | $a_{i0}$ | $a_{i1}$ | $a_{ik}$ | $X_i$ | $S_i$  | $T_i$  | $G_i$  | $A_i$   | $D_i$   | $r_i$ |
|----|-------|--------|-------|----------|----------|----------|-------|--------|--------|--------|---------|---------|-------|
| -1 | 20,00 | 400,00 | 50,00 | 0,8      | 0,1      | 0,1      | 17,5  | -      | 750,00 | -      | -0,125  | 20      | 0,05  |
| 0  | 20,00 | 400,00 | 50,00 | 0,8      | 0,1      | 0,1      | 17,5  | -      | 750,00 | -      | -0,125  | 20      | 0,05  |
| 1  | 20,00 | 400,00 | 50,00 | 0,8      | 0,1      | 0,1      | 17,8  | 887,50 | 756,25 | 131,25 | -0,125  | 20      | 0,05  |
| 2  | 18,00 | 398,00 | 51,00 | 0,7      | 0,2      | 0,1      | 17,4  | 890,46 | 710,36 | 180,09 | -0,2881 | 22,4859 | 0,1   |
| 3  | 17,00 | 397,00 | 51,50 | 0,8      | 0,1      | 0,1      | 14,6  | 780,91 | 644,49 | 136,42 | -0,1064 | 16,405  | 0,05  |
| 4  | 16,00 | 396,00 | 52,00 | 0,8      | 0,1      | 0,1      | 14,0  | 747,93 | 619,84 | 128,09 | -0,1016 | 15,4688 | 0,05  |
| 5  | 15,00 | 395,00 | 52,50 | 0,8      | 0,1      | 0,1      | 13,3  | 707,20 | 594,04 | 113,16 | -0,0972 | 14,6296 | 0,05  |
| 6  | 14,00 | 394,00 | 53,00 | 0,8      | 0,1      | 0,1      | 12,6  | 680,20 | 570,89 | 109,31 | -0,0933 | 13,8732 | 0,05  |
| 7  | 13,00 | 393,00 | 53,50 | 0,8      | 0,1      | 0,1      | 12,1  | 654,49 | 549,70 | 104,79 | -0,0898 | 13,1879 | 0,05  |
| 8  | 12,00 | 392,00 | 54,00 | 0,8      | 0,1      | 0,1      | 11,5  | 631,03 | 530,25 | 100,77 | -0,0865 | 12,5641 | 0,05  |
| 9  | 11,00 | 391,00 | 54,50 | 0,8      | 0,1      | 0,1      | 11,0  | 609,43 | 512,34 | 97,09  | -0,0836 | 11,9939 | 0,05  |
| 10 | 10,00 | 390,00 | 55,00 | 0,8      | 0,1      | 0,1      | 10,6  | 589,48 | 495,78 | 93,70  | -0,0809 | 11,4706 | 0,05  |

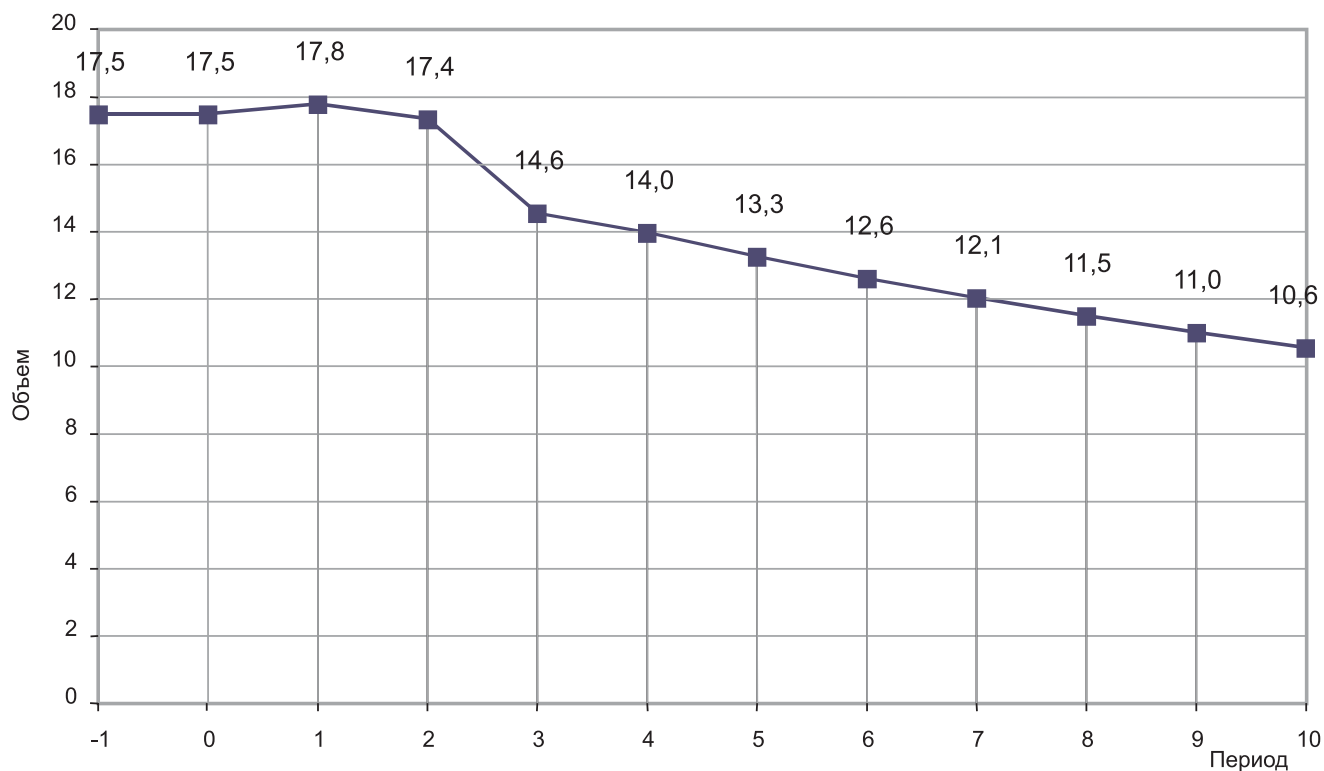


Рис.9.4.2 График объемов выпускаемой продукции

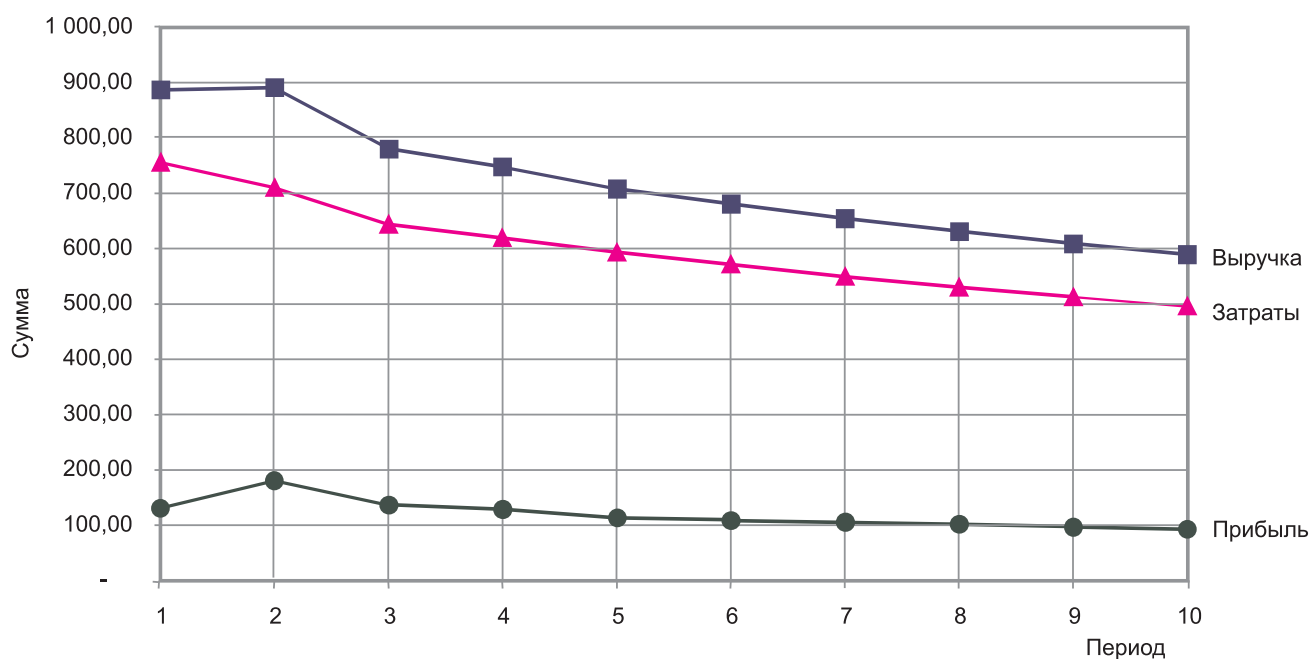


Рис. 9.4.3 Графики изменения величин выручки, затрат и прибыли

$$Z_i = C_{ei}Q_{ei} + C_{fi}Q_{fi} + C_{di} + C_{ri} + \\ + C_{ti} + C_{gi} + C_{ai} \left( 1 + \sum_{j=1}^4 \beta_j \right) + C_{pi}, \quad (15)$$

где

$Q_{ei}$  – расход электроэнергии на непроизводственные цели в  $i$ -й период;

$Q_{fi}$  – расход топлива на непроизводственные цели в  $i$ -й период;

$C_{di}$  – стоимость погашения кредитов (займов) в  $i$ -й период, взятых в предшествующие периоды;

$C_{ri}$  – арендная плата в  $i$ -й период;

$C_{ti}$  – плата за телефонное обслуживание в  $i$ -й период;

$C_{gi}$  – стоимость охраны в  $i$ -й период;

$C_{ai}$  – стоимость зарплаты обслуживающего персонала в  $i$ -й период;

$C_{pi}$  – стоимость амортизационных отчислений на оборудование, стоимость мероприятий по маркетингу и другие постоянные платежи в  $i$ -й период.

При выборе символьных индексов в перечисленных обозначениях мы учитывали первые буквы следующих слов: quantity (количество), warehouse (склад), energy (энергия), salary (зарплата), fuel (топливо), rent (арендная плата), borrow (взять в займы), guard (охрана).

Для практического использования этих формул в процессе имитационного моделирования достаточно построить ARIS-модели продукции, в которых описать состав продукции и затраты на приобретение комплектующих изделий, материалов и т. д. В качестве примера на рис.9.5.1, 9.5.2 описаны затраты на производство мужских сорочек. Атрибуты объектов этих моделей содержат всю необходимую информацию для применения формул (14), (15). В 7-й главе представлена ARIS-программа вычисления переменных и постоянных затрат для производства продукции (необязательно сорочек), модель которой описывается диаграммами, примеры которых представлены на рис.9.5.1, 9.5.2.

Следует отметить еще одно обстоятельство. Рассмотренные модели затрат являются в определенной степени упрощенными, поскольку не содержат таких платежей, как налог на добавленную стоимость, налог на прибыль, налог на пользователей автодорог и т. д. Учет перечисленных налогов приведет к соответствующему усложнению приведенных формул. Корректировка формул (14), (15) с целью учета налога на добавленную стоимость приведена, например, в работе [11].

## 9.6. Сценарий моделирования финансово-обеспеченного производства

Кроме резервных средств для покрытия расходов на производство, привлекаются средства, вырученные от реализации продукции, произведенной в расчетный и предшествующие периоды. Общий вид формулы вычисления выручки содержит объемы произведенной продукции, цену продукции в расчетный период, число периодов реализации продукции. Эта формула описывает влияние на цену продукции времени ее хранения и другие факторы. Величина цены продукции является параметром, который связывает БУ с моделями спроса и предложения. Если рыночная цена равна или выше принятой в БУ, то нет необходимости дополнять это БУ моделями спроса и предложения. Если цена продукции в БУ является исковой, то ее можно выразить через остальные величины.

Параметр, описывающий число периодов реализации продукции, характеризует работу маркетинговых служб. Идеальная работа этих служб обеспечивает реализацию продукции по завершению ее производства (без необходимости складирования и содержания запасов). Но реально произведенная продукция в текущий период реализуется не только в этот, но и в последующие периоды. Причем выручка, полученная в текущий период от реализованной продукции, и собственные финансовые средства могут не покрыть затраты на производство продукции в этот период. В этом случае необходимы заимствования.

Для покрытия расходов (левой части БУ) в расчетный (моделируемый) период могут привлекаться заемные средства или продаваться собственность (активы) предприятия. Остановимся подробнее на заимствовании средств в банке.

Заемные средства разобьем на две части. Первая из них считается взятой до процесса моделирования, например, для начала бизнеса или для приобретения материалов, комплектующих изделий, энергоресурсов и т. д. Последствия этой операции отражаются в затратной части БУ в виде слагаемого, которое описывает поток платежей (в банк) в каждый расчетный период за взятый до начала моделирования кредит. Вторая часть заимствований относится к расчетному периоду и стоит в правой (приходной) части БУ. Она описывает сумму кредита, которая необходима для сбалансирования левой части БУ, а также, возможно, для пополнения резервных средств, запасов, для воспроизводства средств производства и т. д.

Так как резервные средства предприятия, как правило, хранятся в банках и создают прибыль предприятию, то для оптимизации финансовой деятельности предприятия можно дополнить БУ соответствующими



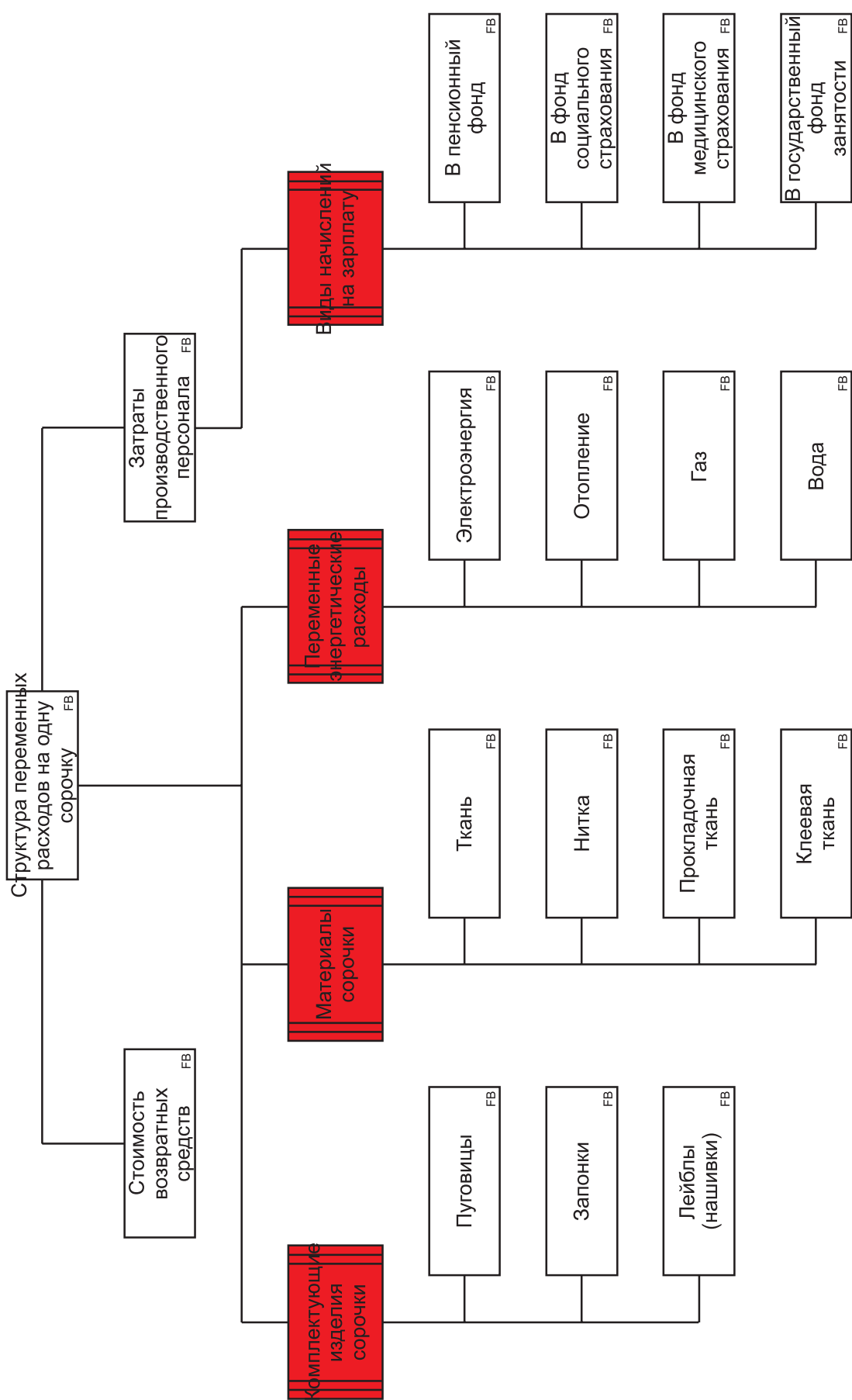


Рис.9.5.1. Диаграмма технических терминов для понятий структуры переменных расходов на производство одной сорочки

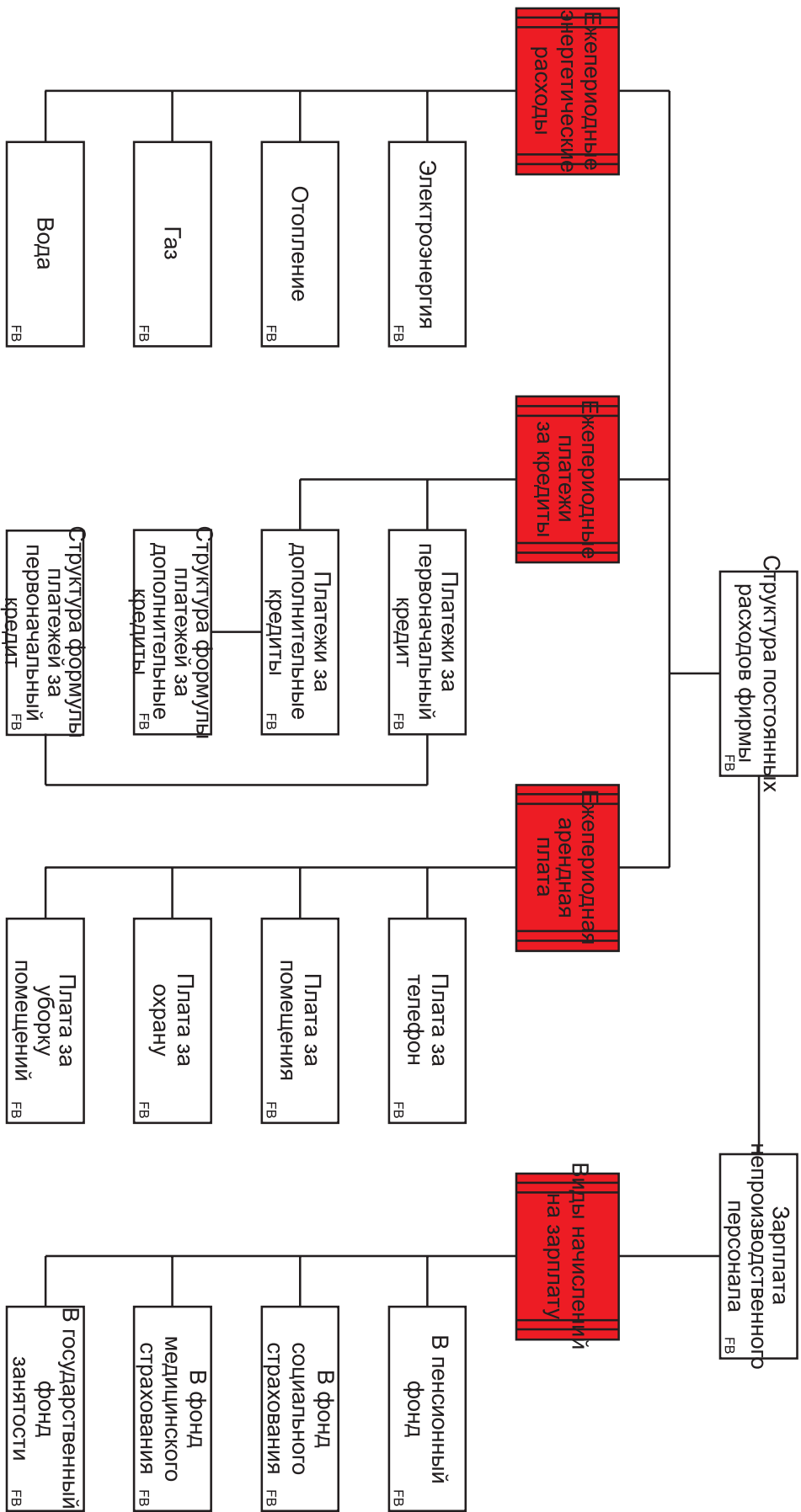


Рис.9.5.2. Диаграмма технических терминов для понятий структуры постоянных расходов на производство одной сорочки

уравнениями из финансовой математики, которые описывают прямые и обратные финансовые потоки по отношению к предприятию. Здесь под прямым потоком (от предприятия в банк) подразумевается аккумуляция финансовых средств предприятия в банках под проценты с соответствующим получением прибыли, под обратным – взятие кредитов в банке с соответствующей платой за них. Опишем сценарий моделирования финансово-обеспеченного производства с учетом прямых и обратных финансовых потоков.

В БУ величина  $Z_i$  содержит слагаемое  $C_{di}$  – стоимость погашения кредитов в  $i$ -й период, взятых в предшествующие периоды. Величина  $C_{di}$  вычисляется по формуле  $C_{di} = C_{di}^a + C_{di}^b$ , где  $C_{di}^a$  – выплата в  $i$ -й период за кредит, взятый до начала моделирования,  $C_{di}^b$  – выплата в  $i$ -й период за кредиты, взятые в процессе моделирования. Величина  $C_{di}^b$  может появиться в левой части БУ в любой период после начала моделирования. Если же предприятие компенсирует свои затраты только за счет выручки и резервных средств, а затраты на обслуживание дополнительного кредита больше прибыли от собственных средств в банке, то величина  $C_{di}^b$  всегда равна нулю. Для вычисления величин  $C_{di}^a$ ,  $C_{di}^b$  и  $C_i^r$  необходимо составить три расчетные формулы и добавить их к БУ.

Формула для вычисления  $C_{di}^a$  содержит заданную величину первоначального займа, срок моделирования и процентную ставку, которая может измениться в любой из расчетных периодов, если, например, по условиям договора она привязана к ставке рефинансирования центрального банка России (ЦБ). Поэтому нельзя заранее (до начала моделирования) вычислить ежепериодные выплаты  $C_{di}^a$  по основному кредиту. Здесь  $i=1,2, \dots, N_0$ ,  $N_0$  – число периодов моделирования.

Формула для вычисления величины  $C_{di}^b$  модифицируется в тот период, когда возникает необходимость взять дополнительный кредит. В этой формуле добавляется величина платежа за очередной дополнительный кредит, рассчитанный в зависимости от величины  $C_i^d$ , номера периода  $i$  и процента, который также зависит от  $i$ . Здесь считается, что дополнительный кредит можно взять в любой расчетный период, но погасить его необходимо до конца моделирования ежепериодными выплатами, где процентная ставка привязана к ставке рефинансирования ЦБ и начисляется на оставшуюся сумму долга. В противном случае последствия взятия дополнительного кредита не будут учтены за время моделирования.

Формула вычисления  $C_i^r$  зависит от расчетного значения и определяется стратегией финансовой дея-

тельности предприятия. Мы ее рассмотрим при описании последовательности и логики моделирования.

Таким образом, при моделировании расчеты будут вестись по четырем формулам. Будем считать, что величины  $U_i$ ,  $Z_i$ ,  $C_i$  известны или вычисляются по соответствующим расчетным формулам, например,  $U_i$ ,  $Z_i$  по формулам (14,15), а цена определяется по модели спроса и предложения, т. е. совпадает с рыночной. Поэтому БУ будем использовать для вычисления  $X_i$  или  $C_i^d$ .

Процесс моделирования в  $i$ -й период будем осуществлять по следующему сценарию.

1. Вычислить величины  $U_i, Z_i$  (в  $Z_i$  учесть платежи за ранее взятые кредиты).
2. Определить (задать) величину  $C_i$ .
3. Вычислить  $X_i$  из БУ при  $C_i^d = C_i^r = 0$ , т. е. положить  $X_i = (S_i - Z_i) / U_i$ .
4. Если  $|X_i - X_{i-1}| \leq \Delta X$ , где  $\Delta X$  – допустимый уровень отклонения объемов производства от периода к периоду, то вычисленный объем производства является финансово-обеспеченным за счет выручки. Поэтому можно перейти к предпоследнему шагу (7).

5. Если  $X_i > X_{i-1} + \Delta X$ , т. е. выручка от реализации продукции позволяет произвести в  $i$ -й период продукции больше допустимой величины, то часть этой выручки направить на увеличение резервных средств. Для этого вычислить максимально допустимый объем производства по формуле  $X_i = X_{i-1} + \Delta X$ , затем вычислить прибавку  $\Delta C_i^r$  к резервным средствам и ее новое значение по формулам

$\Delta C_i^r = S_i - U_i X_i - Z_i$ ,  $C_{\max}^r = C_{\max}^r + \Delta C_i^r$ , где  $C_{\max}^r$  – максимальные резервные средства предприятия.

6. Если  $X_i < X_{i-1} + \Delta X$ , т. е. выручки недостаточно для финансового обеспечения выпуска продукции в  $i$ -й период не ниже допустимого уровня, то необходимо привлечь собственные средства, и, если их недостаточно, взять дополнительный кредит.

6.1 Вычислить размер дополнительных финансовых средств  $C_d$  для обеспечения объема производства на достигнутом уровне, т. е. положив  $X_i = X_{i-1}$ , вычислить величину выручки  $S_i$  и найти величину  $C_d$  по формуле

$$C_d = U_i X_{i-1} + Z_i - S_i.$$

6.2 Если  $C_d \leq C_{\max}^r$ , т. е. величина  $C_d$  не выше максимальных резервных средств, то в  $i$ -й период для сбалансированности затрат необходимо привлечь из резервных средств сумму  $C_i^r = C_d$  и уменьшить  $C_{\max}^r$  на величину  $C_d$ , т. е. положить  $C_{\max}^r = C_{\max}^r - C_d$ .

6.3 Если  $C_d > C_{\max}^r$ , т. е. величина  $C_d$  больше всех оставшихся у предприятия резервных средств,

то в  $i$ -й период для сбалансированности затрат необходимо привлечь все резервные средства (положить  $C_{di}^r = C_{\max}^r$ ) и взять дополнительный кредит в сумме  $C_{di}^d = U_i X_i + Z_i - S_i - C_{\max}^r$ .

Так как все резервные средства на этом этапе использованы, то положить  $C_{\max}^r = 0$ .

7. Вывести результаты моделирования на  $i$ -м шаге. Перейти к следующему периоду, т. е. положить  $i = i + 1$ . Если  $i \leq N_0$ , то продолжить процесс моделирования с первого шага. Если  $i > N_0$ , то перейти к следующему (последнему) шагу.

8. Завершить моделирование, сохранив его результаты для анализа.

Рассмотрим формулы вычисления величин  $C_{di}^a$ ,  $C_{di}^b$ . Пусть  $P_1$  – величина первоначального (основного) кредита, взятого до начала моделирования на  $N_0$  периодов с условием ежепериодного погашения. Таким образом, в каждый период гасится процент за пользование кредитом и часть кредита в сумме  $P_1/N_0$ , которая не меняется от периода к периоду. Здесь  $N_0$  – количество периодов моделирования. Проценты начисляются по ставке  $i_1 = j_1/m$  на оставшуюся сумму кредита, которая уменьшается от периода к периоду. Здесь  $j_1$  – номинальная годовая процентная ставка,  $m$  – число начислений процентов в году (число периодов конверсии). Таким образом,  $i_1$  – простая процентная ставка применительно к одному периоду конверсии.

Для выполнения условий кредитного договора составляется схема погашения кредита с процентами. Схема погашения основного кредита предполагает, во-первых, ежепериодную выплату части кредита в сумме  $P_1/N_0$ , во-вторых, поток платежей выплат процентов за пользование кредитом, который описывается следующей последовательностью:

$$P_1 i_1, \frac{P_1(N_0 - 1)}{N_0} i_1, \frac{P_1(N_0 - 2)}{N_0} i_1, \\ \frac{P_1(N_0 - 3)}{N_0} i_1, \dots, \frac{P_1[N_0 - (N_0 - 1)]}{N_0} i_1 = \frac{P_1}{N_0} i_1,$$

где первый член равен сумме выплачиваемого процента за первый период, второй равен выплачиваемому проценту за второй период и т. д. до последнего периода.

Начиная с  $k$ -го периода конверсии от начала моделирования, фирма берет дополнительный кредит в связи с производственной необходимостью. Сумма второго кредита  $P_2$ , процентная ставка  $i_2 = j_2/m$  (понятия величин  $i_2$ ,  $j_2$  те же, что и в первом случае). Составляется аналогичная схема погашения. Оба кредита должны быть погашены к концу срока моделиро-

вания. Схема погашения дополнительного кредита предполагает, во-первых, ежепериодную выплату части кредита в сумме  $P_2/(N_0 - k)$ , во-вторых, поток выплат процентов за дополнительный кредит, который описывается последовательностью

$$P_2 i_2, \frac{P_2}{N_0 - k} (N_0 - k - 1) i_2, \\ \frac{P_2}{N_0 - k} (N_0 - k - 2) i_2, \dots, \frac{P_2}{N_0 - k} i_2,$$

где первый член равен сумме выплачиваемого процента за  $k$ -й период, второй – за  $(k + 1)$ -й период и т. д., последний член равен процентному платежу за последний период моделирования.

Для осуществления описанного моделирования вся входная информация, планируемые задачи и их исполнители должны описываться в функциональной, организационной и информационной ARIS-моделях, которые составляют пользователем для конкретной производственно-сбытовой фирмы. События, последовательность и логика решаемых задач должны описываться в моделях процессного вида. Такой подход позволяет не только обеспечить входной информацией математические модели, но и создать исчерпывающую информационную базу для выполнения имитационного моделирования.

### 9.7. Моделирование фирмы по производству и реализации мужских сорочек

Рассмотрим вымышленную (гипотетическую) фирму, занимающуюся производством и реализацией (продажей) мужских сорочек. Диаграмма организационной структуры этой фирмы изображена на рис.9.7.1, из которой видно, что на фирме работает восемнадцать человек, из них двое занимают (разделяют) должность заместителя директора, Фаттахова О. В. совмещает должности раскройщицы и портнихи, а Кондратьев А. С. совмещает должность второго продавца и кладовщика. Господин Мороз А. В. по отношению к фирме является внешней персоной. Эти и многие другие особенности всех субъектов ответственности (персон) фирмы можно узнать из их атрибутов. Например, из атрибутов Address и Description объекта типа Person с именем Мороз А. В. можно узнать, что господин Мороз А. В. проживает в г. Екатеринбурге и работает на фирме по договорам. Должностные инструкции каждого сотрудника содержатся в отдельном Word-документе и через атрибуты Title 1, Link 1 и Parameter 1 связаны с

объектом типа Person. Фотографию этих персон можно посмотреть, если инициализировать бинарный код фотографии, связанный с объектом Person через атрибуты Title 2, Link 2 и Parameter 2. В качестве примера на рис.9.7.2 распечатаны атрибуты Description и фотографии четырех сотрудников моделируемой фирмы.

Фирма занимается производством и реализацией мужских сорочек. Из диаграмм на рис.9.5.1, 9.5.2 видно, что для этого она арендует помещения, платит за пользование телефоном, за уборку занимаемых помещений и т. д. Каждая сорочка комплектуется пуговицами, запонками, нашивками (лейблы) и т. д. Атрибуты объектов этих моделей содержат все необходимые данные для вычисления переменных и постоянных затрат в каждый расчетный период.

Цены на комплектующие изделия и материалы зависят от поведения поставщиков и работы группы снабжения, т. е. являются следствием имитации БП снабжения. При стабильной работе группы снабжения они не меняются, а при сбоях в поставках из-за изменившихся внешних условий они могут возрасти. Аналогично могут меняться атрибуты других объектов. Значения этих атрибутов используются для вычислений  $U_i$ ,  $Z_i$  по формулам (14), (15), структура которых описана в диаграммах на рис.9.5.1, 9.5.2.

На рис.9.7.3 представлена диаграмма основной деятельности рассматриваемой фирмы. Здесь мы детализируем только те БП, которые влияют на формирование входных, промежуточных и результатных данных. Для уменьшения громоздкости описания и экономии места приведем модели БП, в которых участвуют только сотрудники производственного отдела и магазина. Деятельность этих сотрудников отражена в диаграммах, представленных на рис.9.7.4, 9.7.5.

На рис.9.7.5 изображен процесс обслуживания клиента в магазине. Если этот процесс протекает по вертикали, то временные затраты на обслуживание клиента – минимальны. Если же процесс протекает по левым или правым ветвям в тех или иных комбинациях, то затраты на обслуживание возрастают. Алгоритм увеличения затрат отражается в миниспецификациях, которые связаны с логическими операторами и событиями, переключающими процесс с магистрали, т. е. с вертикального пути развития. Напомним, что магистралью в математической экономике называют оптимальный путь развития (в смысле минимизации затрат) экономического процесса. Проход по тому или иному пути определяется (задается) пользователем модели при анимации в режиме ручного выбора пути развития процесса в точках его разветвления. Эти же замечания относятся и к возможным путям

развития БП на рис.9.7.4. Очевидно, что вертикаль этого БП является магистральной. Действительно, если сорочка после изготовления сразу прошла контроль качества, то это идеальное производство.

Заметим, что сойти с магистрали можно только на логическом операторе с одним входом и несколькими выходами. Это происходит на операторе “исключающее или”, реже – на операторе “или”, но никогда – на логическом операторе “и” и на операторе “или”, от которого все простые ветви замыкаются на операторе “и”.

Построенные модели и выведенные расчетные формулы позволяют осуществить дискретное динамическое моделирование деятельности рассматриваемой фирмы.

## 9.8. Особенности дискретного динамического моделирования

Для большинства фирм непрерывное получение и обработка количественных данных о производстве продукции, динамике продаж, накоплении и расходовании финансовых средств является ненужным или затруднительным ввиду недостаточного внедрения автоматизированных систем получения и обработки информации. Поэтому на практике обработка данных о ходе производства, затратах, платежах за кредиты и поступлении выручки проводится дискретно с постоянным периодом (сутки, неделя, месяц, квартал и т. д.). По этой причине рассмотренные динамические модели являются дискретными, в которых номера периодов дискретности обозначены символами  $i$ ,  $(i-1)$ ,  $(i-2)$ , ...,  $(i-k)$ . При этом символом  $i$  обозначается рассматриваемый (текущий, расчетный) период. Символом  $(i-1)$  обозначается период, предшествующий  $i$ -му. Символом  $(i-k)$  обозначается период, отстоящий от  $i$ -го на  $k$  периодов.

Использование дискретных динамических моделей дает возможность прогнозировать наступление моментов дестабилизации деятельности предприятия и принимать заблаговременно решения по их недопущению. Определение объемов производства  $X_i$  позволяет заранее готовить варианты разумных решений на прогнозируемые изменения цен ресурсов, корректировать планы производства продукции, получения кредитов, изменения отпускных цен.

При некомпенсированности затрат и отсутствии возможности привлечь дополнительные финансовые ресурсы плановый объем производства должен быть скорректирован в сторону уменьшения или должны быть приняты меры по уменьшению затрат, так как иначе возникнет проблема неплатежеспособности. Моделирование позволяет прогнозировать моменты





Долгих Александр Александрович  
Адрес: г. Миасс, ул. Молодежная, д. 14, кв. 35  
Телефон: 2-20-45  
E-mail: dol@inf.susu.ac.ru  
Должность: Заместитель директора  
(на 0,5 ставки)



Максимов Владимир Иванович  
Адрес: г. Миасс, ул. Макеева, д.7 кв. 42  
Телефон: 2-72-80  
E-mail: maks@inf.susi.ac.ru  
Должность: Заместитель директора  
(на 0,5 ставки)



Михайлов Александр  
Адрес: г. Миасс, пр. Октября, д. 51, кв. 77  
Телефон: 2-56-53  
E-mail: almi@inf.susu.ac.ru  
Должность: Специалист по снабжению



Коваль Милена Леонидовна  
Адрес: г. Миасс, ул. Менделеева, д. 5, кв. 12  
Телефон: 4-47-48  
E-mail: milena@inf.susu.ac.ru  
Должность: Главная раскройщица



Рис.9.7.2 Выборочный отчет по диаграмме на рис.9.7.1





Рис.9.7.3. Диаграмма цепочек основных процессов производственно-сбытовой фирмы

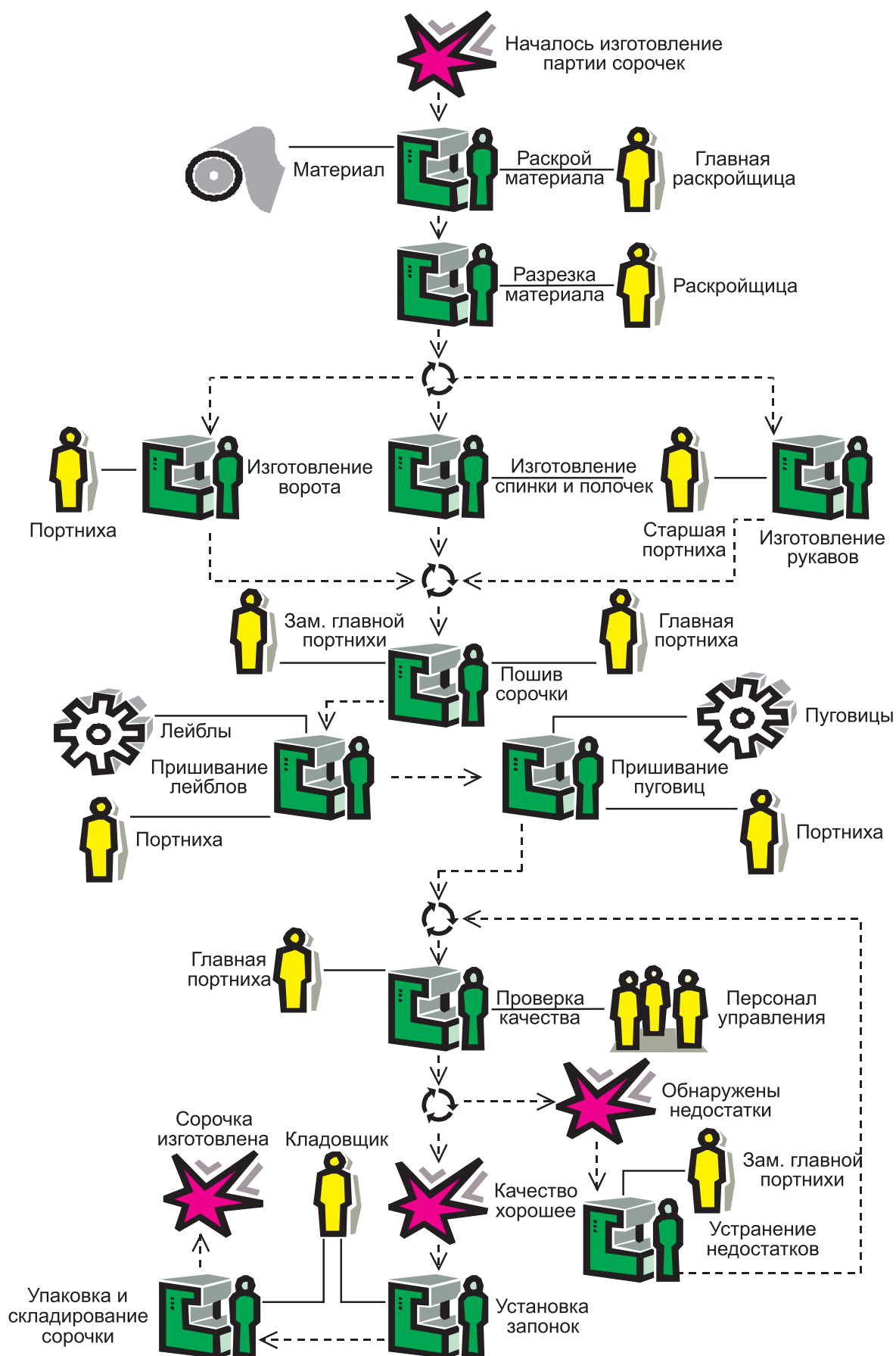


Рис.9.7.4 Диаграмма процесса изготовления партии сорочек

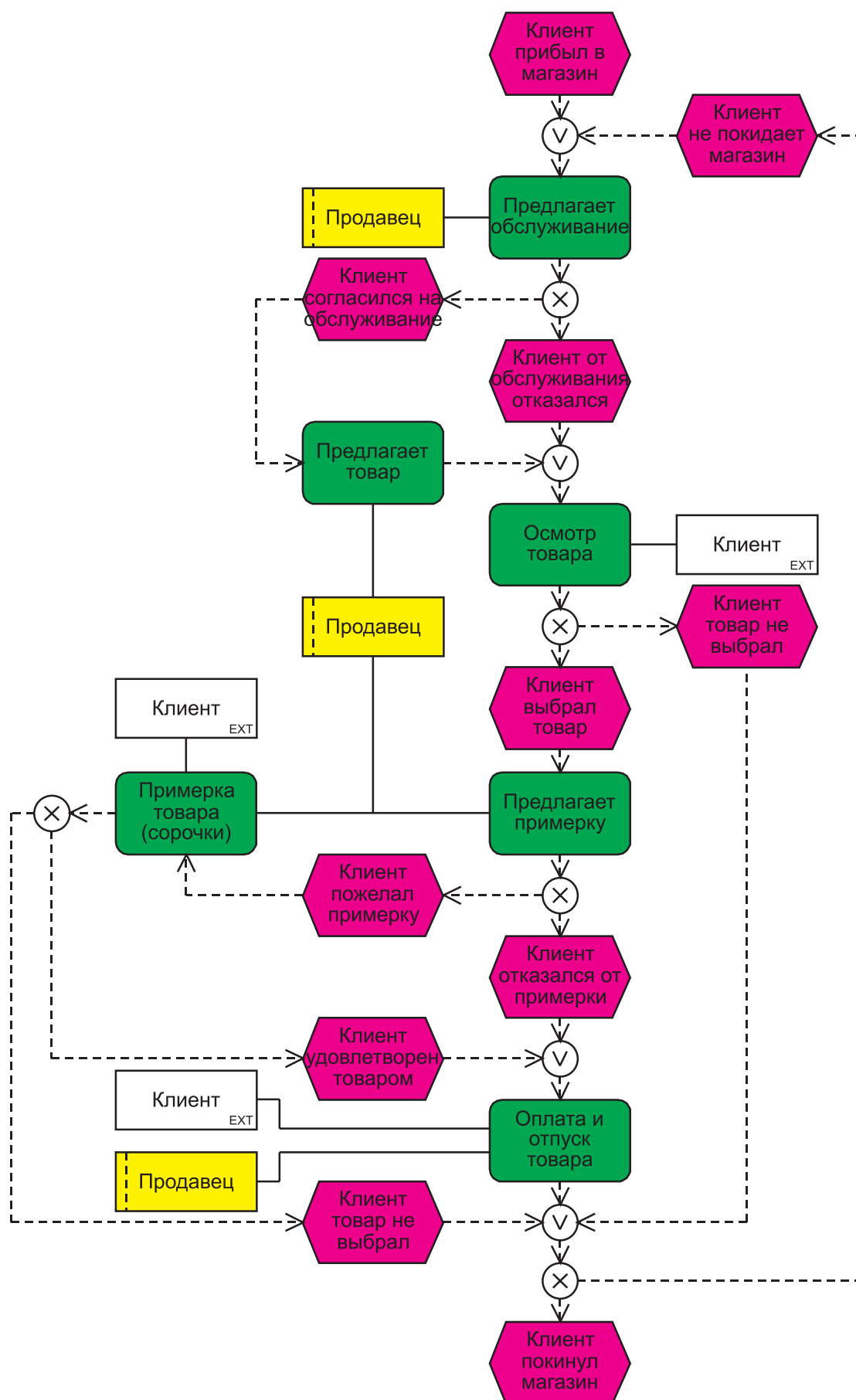


Рис.9.7.5. Процесс обслуживания клиента в магазине

наступления критических для предприятия ситуаций, выявлять причины их возникновения и вовремя принимать решения по корректировке процессов производственной и сбытовой деятельности.

Путем сопоставления динамики производства товаров с динамикой их реализации руководство предприятия планирует объемы производства в будущие периоды и разрабатывает мероприятия по повышению прибыльности предприятия, по ускорению сбыта и продвижению продукции на рынки потребителей.

Результаты анализа БП предприятия готовятся на конец расчетного периода, так как основные поступления за реализацию продукции, а тем более фактические затраты (заработная плата, плата за электроэнергию, топливо, сырье, материалы, комплектующие изделия, платежи в пенсионный фонд, фонд социального страхования, фонды медицинского страхования, государственный фонд занятости населения, налог на прибыль и другие платежи) осуществляются в конце расчетного периода.

Для выработки управленческих решений на основе описанных моделей в реальном масштабе времени необходима информация об имеющихся собственных  $C_i^r$  и привлекаемых  $C_i^d$  финансовых средствах, об объеме выпущенной продукции в предыдущие периоды и удельном весе ее реализации в последующие периоды, о рыночной цене на продукцию, обязательных налогах, платежах и других величинах, входящих в переменные  $U_i$  и постоянные  $Z_i$  затраты. Учет этих сведений должен вестись бухгалтерией предприятия. Это позволит быстро готовить исходные данные для моделирования, проводить расчеты и вырабатывать информацию, необходимую для принятия решений по управлению предприятием.

Для поддержки исходных данных в актуальном состоянии необходимо ежепериодно обновлять их, например, в случае изменения стоимостных показателей сырья, материалов, комплектующих изделий, электроэнергии, топлива, заработной платы, налогов, обязательных платежей. Необходимую информацию по исходным данным целесообразно сгруппировать по изделиям или по другому принципу и составить соответствующие иерархические ARIS-модели.

Особенностью рассмотренных динамических моделей является наличие обратной связи. Входящие в них слагаемые  $X_{i-1}, X_{i-2}, X_{i-3}, \dots, X_{i-k}$  являются входной информацией для  $i$ -го периода, но представляют выходную информацию (объемы производства) соответственно в период  $(i-1), (i-2), (i-3), \dots, (i-k)$ . Таким образом, выходная информация в  $i$ -й период задерживается на один, два, три, ...,  $k$  периодов и по

обратной связи влияет на информацию следующих за ним периодов. Положительная обратная связь действует с запаздыванием на один, два, три, ...,  $k$  периодов. При прогнозировании изменений экономических показателей предприятия моделируется ряд следующих друг за другом периодов, выходные показатели которых сохраняются в памяти компьютера в виде величин  $X_i, X_{i-1}, X_{i-2}, X_{i-3}, \dots, X_{i-k}$  и других показателей. Эта особенность моделей должна учитываться при разработке программ моделирования изучаемых экономических процессов. Результаты динамического моделирования для основных показателей можно предоставлять пользователю в конце каждого расчетного периода и запоминать для дальнейшей работы или просмотра.

В зависимости от целей моделирования и степени оснащенности предприятия вычислительной техникой, могут быть применены различные способы использования описанных моделей: от выполнения несложных расчетов по конечным формулам для получения экономических показателей предприятия в рассматриваемый период до разработки общей комплексной динамической модели с единой БД и компонентами для расчета отдельных показателей функционирования предприятия за несколько периодов.

Большинство исходных данных, необходимых для проведения экономического анализа, являются данными бухгалтерского учета. Поэтому все исходные данные целесообразно готовить в бухгалтерии по удобным для них формам. Специальный конвертер (ARIS-программа) перенесет эти данные в соответствующие атрибуты ARIS-диаграмм, описывающих информационную модель анализируемой (управляемой) системы.

Повысить эффективность деятельности предприятия можно за счет уменьшения постоянных и переменных затрат. Величины  $Z_i, U_i$  можно уменьшить путем инвестиций  $I$ , т. е. функции  $Z_i(I), U_i(I)$  могут убывать с течением времени, если инвестиции  $I$  направлены на снижение издержек. Рассмотрим некоторые аспекты моделирования инвестиционного процесса.

Инвестиции по направлению вложений денежных средств делятся на чисто финансовые операции (выдача кредитов, покупка ценных бумаг, инвестиции в различные финансовые инструменты и т. д.) и производственные инвестиции. Под последними понимают вложения денежных средств в создание, реконструкцию или переоборудование производственных предприятий. Оба вида инвестиций с финансовой точки зрения охватывают этап вложения денежных средств и этап отдачи от такого вложения. Объектом анализа может быть один

из этих этапов или оба вместе. Информационной базой для анализа указанных этапов являются соответствующие потоки платежей.

При определении эффективности инвестиций необходимо учитывать специфику распределения затрат и доходов во времени. Долгосрочный характер производственных инвестиций многократно усиливает важность применения принципа неравноценности денег, относящихся к разным моментам времени, т. е. зависимости ценности денег от времени.

Для анализа сложных инвестиционных схем разрабатывают специальные экономико-математические модели, которые позволяют получить варианты поведения исследуемого инвестиционного процесса для разнообразных сочетаний исходных условий и принятых предположений, например, состояния денежно-кредитного рынка, уровня инфляции, спроса на выпускаемую продукцию и т. д. Особенность модели, разрабатываемой для инвестиций в производство, состоит в том, что в ней базовым является блок формирования затрат и отдачи от инвестиций для каждого периода со специфическим их распределением. В отдельном блоке модели вычисляются показатели эффективности.

Модель дает возможность осуществить анализ чувствительности. Этот анализ заключается в выявлении наиболее важных (ключевых) входных параметров модели и получении системы оценок эффективности инвестиций для широкого диапазона значений этих параметров. Таким образом, лицу, принимающему решение, предоставляется не единственная, точечная оценка, а развернутая картина (в виде таблиц и графиков) значений эффективности для разнообразных возможностей и ожидаемых ситуаций.

Первый шаг при разработке базового блока модели заключается в определении потока ежепериодных платежей. Причем, и затраты, и доходы в модели должны быть увязаны как с внешними условиями, так и с производственными параметрами, например, в связи с ожидаемой динамикой цен на производимую продукцию (внешние условия) и возможными изменениями объемов производства и уровней текущих производственных затрат. Данные о затратах и доходах в зависимости от конкурентных условий могут быть постоянными и

переменными. Доходы представляют собой дискретный поток поступлений.

В модели выделяют несколько видов данных, например, уровневые или объемные характеристики (выпуск продукции, затраты на строительство и т. д.), временные параметры (продолжительность расчетного периода и общего срока инвестирования, число периодов моделирования), нормативные показатели (процентные и налоговые ставки, удельные расходы и др.).

Зависимость потоков затрат и поступлений от множества данных, относящихся к будущему, не позволяет получить однозначные ответы о степени эффективности, так как цены на продукцию могут понизиться, а затраты – возрасти и т. д. Практически полезно для сокращения риска в условиях неопределенности получить крайние оценки, т. е. применить сценарный подход и получить три оценки. Первую – для базового варианта исходных данных и предпосылок, сформулированных для наиболее вероятного функционирования предприятия. Две другие оценки получить для пессимистичного и оптимистичного вариантов условий протекания БП. Совокупность таких расчетных оценок дает возможность более полно представить финансовые последствия деятельности предприятия, в том числе – инвестиционной.

Более информативным является анализ чувствительности БП. Речь идет об отзывчивости показателей эффективности БП на изменения данных в базовом варианте условий. Выделяют следующие этапы при осуществлении анализа чувствительности. Во-первых, выбор показателя эффективности, относительно которого проверяется отзывчивость системы на изменение того или иного параметра базового варианта условий. Во-вторых, отбор ключевых переменных модели, т. е. данных, отклонения значений которых от базовых заметно отразятся на величине показателя эффективности. В итоге показатель эффективности определяется как функция ключевых переменных модели. Остальные переменные рассматриваются в модели как константы. В-третьих, определение ожидаемых диапазонов значений ключевых переменных. В-четвертых, расчет значений показателей эффективности для принятых диапазонов ключевых переменных и представление результатов расчетов в удобной для анализа форме.

## ***ЗАКЛЮЧЕНИЕ***

В условиях глобализации экономики, роста конкуренции, лавинообразного потока новых информационных технологий, активного перехода на процессное управление, а также других бурных экономических и технологических перемен предприятия вынуждены быстро адаптироваться и перестраивать свои БП под изменившиеся рыночные условия с целью сохранения конкурентоспособности. Как это делать более эффективно, изложено в этой книге. При ее написании, подборе и переработке материала, составлении оригинальных примеров, новых алгоритмов, математических моделей и методик мы придерживались одного принципа – писать о тех предметных областях, которые имеют отношение к бизнес-информатике и для которых строятся конкретные ARIS-диаграммы. Придерживаясь этого принципа, мы привели более ста примеров построения основных типов моделей. В некоторых из них продемонстрирован комплексный подход, т. е. от текстового описания до формального построения моделей с их анализом, расчетами и соответствующими выводами.

Мы стремились на конкретных примерах продемонстрировать ARIS как мощный инструмент наведения порядка в бизнес-деятельности и показать, как с его помощью можно получить материалы для анализа деятельности предприятия с целью осмысления, улучшения и оптимизации этой деятельности. Десятки околоэкономических и игровых примеров демонстрируют, что ARIS – это инструмент для диаграммного описания любых предметных областей, систем и их процессов, что диаграммы позволяют просто выразить сложную концепцию, передать идею понятнее и полнее, чем словесное описание, что графические образы (диаграммы) в ARIS служат базой данных для программ анализа и оптимизации БП.

Мы стремились изложить базовые принципы методологии и основные моменты формализма, которых рекомендуется придерживаться для эффективного использования ARIS. Читая документацию по ARIS, невозможно научиться моделировать бизнес-системы. Изучение только приемов работы с программной системой ARIS этому тоже не поможет. Для освоения бизнес-

моделирования нужно, во-первых, изучить методику формального описания бизнес-деятельности, во-вторых, разобрать конкретные примеры ее применения, и, наконец, в-третьих, научиться выбирать типы ARIS-моделей, правильно структурировать информацию, анализировать, проверять и оформлять модели, устанавливать связи между ними и т. д. Про все это в настоящей книге не только написано, но и продемонстрировано на примерах. К сожалению, полное (исчерпывающее) изложение практически значимых (для реальной бизнес-деятельности) примеров занимает очень много места. Ярких и коротких примеров в литературе очень мало – это золотой фонд примеров и задач для изучения бизнес-моделирования. Надеемся, что мы внесли свой вклад в наполнение этого фонда.

В аннотации и во введении мы перечислили тех, для кого задумывалась эта книга. Сейчас мы видим, что круг ее читателей шире. К нему можно отнести бизнес-менеджеров, менеджеров проектов, торговых представителей, менеджеров по маркетингу, консультантов и специалистов по техническому обслуживанию, финансовых и бизнес-аналитиков, административных и руководящих работников, ИТ-специалистов, разработчиков программного обеспечения и баз данных, управляющих предприятиями, руководителей производств, специалистов по территориальному управлению, ИТ-консультантов.

В настоящее время мы разрабатываем методы ARIS-моделирования роботизированных технологических комплексов, гибких производственных линий и “тонкого” производства, где оптимизируются все процессы (от рассмотрения заказа клиента до поставки ему готовой продукции). Здесь моделируется решение задач стратегического, тактического и исполнительного уровней управления с соответствующими математическими расчетами, анализом и оптимизацией. Основной упор делается на измерение и математическое моделирование БП, разработку оригинальных программ (скриптов) анализа, где ARIS-модели являются информационной базой расчетов.

**ТАБЛИЦА ТИПОВ МОДЕЛЕЙ ARIS 5.0 С ПОЛНЫМ ФИЛЬТРОМ**

| <b>Название вида моделей</b>                   |   |  |  |                |
|--|---|--|--|----------------|
| <b>N</b>                                       | <b>Название<br/>типа модели<br/>(на английском<br/>языке)</b> | <b>Название<br/>типа модели<br/>(на русском языке)</b> | <b>Краткое описание назначения модели</b>  | <b>Фильтры</b> |
| <b>Organization view (Организационный вид)</b> |   |  |  |                |
| 1  | Network diagram   | Диаграмма<br>вычислительной<br>сети                    | На этапе реализации описываются конкретные топологии сети компании, ее узлы и соединения, а также точное местоположение каждой сети, ее узлов и соединений. Регистрирует конкретные компоненты (с указанием лицензий) аппаратного обеспечения и т. д.        | 5              |
| 2  | Network topology  | Топология сети   | На этапе спецификации описывает топологию сети, включая все компоненты сети и связи между ними. Описывает классы и типы сетей, сетевых узлов и соединений, а также типы компонент аппаратного обеспечения и их расположения.                                 | 5              |
| 3  | Organizational chart  | Организационная<br>диаграмма                           | Описывает организационную структуру предприятия с любой степенью детализации.  | 2-8            |
| 4  | Shift calendar  | Календарь смен   | Описывает календарь смен и его связь с исполнителями и ресурсами.  | 2,5            |
| 5  | Technical resources   | Технические<br>средства                                | Иерархически классифицирует технические средства, которые необходимы для преобразования входных материалов для функций в выходные. При этом указываются классы, типы и экземпляры операционных ресурсов, складского оборудования, транспортных систем и т.д. | 5              |
| <b>Function view (Функциональный вид)</b>      |   |  |  |                |
| 1  | Application system<br>diagram                                 | Диаграмма<br>прикладной<br>системы                     | На этапе реализации описывает конкретную диаграмму типа прикладной системы, т.е. экземпляры прикладных систем (с указанием их лицензий) для описанных типов прикладных систем.   | 5              |
| 2  | Application system<br>type diagram                            | Диаграмма типа<br>прикладной системы                   | При спецификации описывает модульную структуру прикладной системы, используемую для выполнения функций БП, а также описывает транзакции, входы, выходы и графические интерфейсы.   | 5,6,8          |



|                               |                                   |   |  |            |
|-------------------------------|-----------------------------------|---|--|------------|
| 3                             | Function tree                     | Дерево функций                              | Описывает иерархические связи между функциями.   | 3-8        |
| 4                             | Objective diagram                 | Диаграмма целей                             | Описывает иерархию целей компании, факторы успеха и достигающие их функции.  | 3,5, 6,8   |
| 5                             | SAP ALE function model            | Модель связей функций с SAP-приложениями    | Описывает связь функций с SAP-приложениями.  | 5          |
| 6                             | SAP applications diagram          | Диаграмма SAP-приложений                    | Описывает модели-прототипы, ориентированные на модули системы управления предприятием SAP R/3.   | 5          |
| 7                             | Y diagram                         | Y-диаграмма.                                | Описывает деятельность компании на самом верхнем уровне  | 5          |
| <b>Data view (Вид данных)</b> |                                   |   |  |            |
| 1                             | Attribute allocation diagram      | Диаграмма окружения атрибутов               | На этапе спецификации описывает атрибутивный состав сущности. Упрощает описание реляционной модели данных конкретного отношения.   | 5          |
| 2                             | Authorization hierarchy           | Диаграмма иерархии разрешений (авторизации) | Описывает взаимосвязи условий авторизации, используемых в диаграмме ролей. Условия авторизации позволяют гарантировать логическую стройность системы моделей и избегать конфликтов.                                    | 5          |
| 3                             | CD Diagram                        | Диаграмма стоимостных драйверов             | Описывает иерархию стоимостных драйверов, т.е. измеряемых величин, которые оценивают стоимость отдельных операций (функций) в БП. Используется в методе ABC.   | 2,5        |
| 4                             | Cost category diagram             | Диаграмма стоимостных категорий             | Описывает иерархию стоимостных категорий, используемых для вычисления стоимости объектов методом ABC. Описывает происхождение стоимости экономических сущностей.   | 2,5        |
| 5                             | DW – Structure diagram            | Диаграмма структуры хранилища данных        | Описывает структуру хранилища данных, в том числе статические отношения данных и их местоположения.  | 3,5        |
| 6                             | eERM                              | Расширенная модель сущность-связь           | Описывает семантическую модель данных рассматриваемой предметной области.  | 1,3,5, 6,8 |
| 7                             | eERM attribute allocation diagram | Диаграмма распределения ERM атрибутов       | Описывает атрибуты для каждого типа сущности и отношения из диаграммы eERM, т.е. моделирует распределение атрибутов по объектам.   | 3,5, 6,8   |
| 8                             | IE Data Model                     | Модель данных IEF                           | Описывает альтернативную ER-модель данных, поддерживаемую инструментом CASE Information Engineering Facility (IEF).  | 5          |
| 9                             | Information carrier diagram       | Диаграмма носителей информации              | Описывает носители информации для ARIS-сущностей, в том числе входные и выходные данные в форме документов, журналов и ARIS-диаграмм. Позволяет описать некоторые требования к процедурам управления проектами в ARIS. | 5          |

|   |   |  |   |       |
|---|---|--|---|-------|
| 10  | Knowledge structure diagram                   | Диаграмма структуры знаний             | Описывает распределение категорий знаний по объектам.   | 5,6   |
| 11  | Material diagram                              | Диаграмма материалов                   | Описывает и иерархически классифицирует типы материалов, принимающих участие в eERP-процессах с материальными потоками.   | 5     |
| 12  | Relations diagram                             | Реляционная диаграмма                  | На этапе спецификации для построения конкретной реляционной БД описывает существующие отношения, атрибуты и их связи с информационными объектами, введенными на этапе определения требований.                     | 5     |
| 13  | SAP SERM (Strutred entity relationship model) | ER-модель для SAP                      | Описывает ER-альтернативную модель данных. Эту модель часто обозначают как SAP-ERM.   | 3,5,7 |
| 14  | SeDaM model (Semantic data model)             | Модель семантики данных SeDaM          | Описывает ER-альтернативную модель данных в нотации BASF AG, т.е. в нотации семантической модели данных SeDaM (Semantic Data Model).  | 5     |
| 15  | System attribute domain                       | Область системного атрибута            | На этапе спецификации описывает типы областей (данные полей) системных атрибутов, т.е. первичные и внешние ключи, обязательные и описательные поля.   | 5     |
| 16  | System attribute                              | Системные атрибуты                     | На этапе спецификации иерархически упорядочивает типы ARIS-сущностей, т.е. событий, технических терминов, функций, информационных носителей и т.д. Модель ориентирована на решение задач экспорта данных из ARIS. | 5     |
| 17  | Table diagram                                 | Диаграмма таблиц                       | На этапе реализации описывает таблицы и поля системы управления реляционными БД.  | 5     |
| 18  | Technical terms model                         | Модель технических терминов            | Описывает систему технических терминов и их взаимосвязей. Через кластеры данных и технические термины устанавливается взаимосвязь с другими объектами представления данных.                                       | 3-6,8 |
| <b>Processes (control) view (Процесный или управляющий вид)</b> |   |  |   |       |
| 1   | Access diagram                                | Диаграмма доступа                      | Описывает связи между типами объектов моделей данных, функций и организаций. Используется на этапе спецификации.  | 5     |
| 2   | Access diagram (physical)                     | Диаграмма доступа на физическом уровне | Описывает диаграмму доступа на физическом уровне, например, конкретные лицензии прикладных систем. Используется на этапе реализации.  | 5     |
| 3   | Authorization map                             | Карта разрешений (авторизации)         | Описывает распределение условий авторизации, используемых на диаграмме ролей, между ОЕ.   | 5     |

|    |  |  |   |          |
|----|--|--|---|----------|
| 4  | Business control diagram                   | Диаграмма управления бизнесом                      | Описывает все потенциальные риски БП и управление ими. Используется, например, для описания стандартных процессов SAP.  | 5        |
| 5  | Class diagram                              | Диаграмма классов                                  | Диаграмма классов UML, описывающая отдельные информационные объекты.  | 5        |
| 6  | Classification diagram                     | Диаграмма классификации                            | Классифицирует функции, привязывая их к различным классам по тому или иному критерию.   | 5        |
| 7  | Communications diagram                     | Диаграмма коммуникации                             | Описывает ОЕ во взаимодействии друг с другом. Группирует БП компании вокруг соответствующих ОЕ.   | 5,7      |
| 8  | Competition model                          | Модель конкуренции                                 | Описывает конкурентную ситуацию, в которой находится компания, указывая, какие клиенты используют те или иные виды продукции предприятия, а также, какие виды продукции поставляются поставщиками и какие продукты (услуги) предлагаются (потенциальными) конкурентами. | 5        |
| 9  | DW – Transformation diagram                | Диаграмма преобразования данных хранилища данных   | Описывает динамические аспекты хранилища данных.  | 3,5      |
| 10 | eBusiness scenario diagram                 | Диаграмма сценариев электронного бизнеса           | Описывает перекрестные процессы между компаниями, в том числе все процессы между двумя бизнес-единицами и создание добавленной стоимости при использовании новых каналов.   | 1,5      |
| 11 | eEPC (extended event driven process chain) | Расширенная событийно-управляемая цепочка процесса | Описывает последовательность и логику выполнения функций БП, указывая при этом исполнителей, информационные носители, входные и выходные объекты и многое другое.   | 1,2, 4-8 |
| 12 | eEPC (column display)                      | eEPC (в виде столбцов)                             | Описывает то же, что и eEPC, но объекты расставляются в столбцах и моделирование ведется сверху вниз.   | 2,5,6    |
| 13 | eEPC (instance)                            | eEPC (уровень экземпляров)                         | Описывает конкретный экземпляр БП, где все конкретизировано, например, время начала процесса, значения всех атрибутов объектов и т.д.   | 5        |
| 14 | eEPC (material flow)                       | eEPC (с материальными потоками)                    | Описывает динамическую цепочку производственных функций, связанных с объектами других представлений, в том числе с входными и выходными материалами и ресурсами (оборудованием, складами, транспортом и т.д.) для работы с ними.  | 2,5      |
| 15 | eEPC (row display)                         | eEPC (в виде строк)                                | Описывает то же, что и eEPC, но объекты расставляются в строках и моделирование ведется слева направо.  | 5        |

|    |  |   |  |               |
|----|--|---|--|---------------|
| 16 | eEPC (table display)                   | eEPC (в виде таблицы)                             | Описывает то же, что и eEPC, но объекты расставляются в строках и столбцах.  | 5             |
| 17 | Event diagram                          | Диаграмма событий                                 | Описывает логическую связь событий и факты изменения состояний информационных объектов.  | 2,5           |
| 18 | Function allocation diagram            | Диаграмма окружения функций                       | Описывает статические отношения между функциями и другими объектами.   | 1,2<br>4-8    |
| 19 | Function allocation diagram (instance) | Диаграмма окружения функций (уровень экземпляров) | Описывает статические отношения между конкретной функцией и другими конкретными объектами.   | 5             |
| 20 | Function/organizational level diagram  | Распределение функций по организационным уровням  | Описывает ответственность OE за выполнение функций, т.е. связывает функции из функционального дерева с их исполнителями (ответственными) из организационной диаграммы.                               | 5             |
| 21 | Industrial process                     | Производственный процесс                          | Наглядно описывает динамические цепочки производственных функций и их взаимосвязи с другими объектами, в том числе с материалами и продуктами. По содержанию аналогичен eEPC-диаграмме.              | 2,4,5,<br>6,8 |
| 22 | Information flow diagram               | Диаграмма информационных потоков                  | Описывает поток информации между функциями. В зависимости от степени детализации рассматриваемых функций информационных объекты могут представлять кластеры данных, типы сущностей или ERM-атрибуты. | 5-8           |
| 23 | Input/Output diagram                   | Диаграмма входа/выхода                            | Описывает входные и выходные данные и информационные носители для функций, преобразующих входную информацию в выходную.  | 5             |
| 24 | Input/Output diagram (invert)          | Диаграмма входа/выхода (инвертированная)          | Описывает то же, что и Диаграмма входа/выхода, но входы и выходы в ней меняются местами.   | 5             |
| 25 | Knowledge map                          | Карта знаний                                      | Описывает различные категории знаний.  | 5,6           |
| 26 | Material flow diagram                  | Диаграмма потока материалов                       | Описывает материальные потоки между функциями, т.е. отображает материалы и типы материалов, которыми обмениваются функции.   | 5             |
| 27 | Office process                         | Офисный процесс                                   | Предназначен для более наглядного графического представления динамических цепочек функций в офисах и их взаимосвязей с объектами других представлений. По содержанию аналогичен eEPC-диаграмме.      | 2,4,5,<br>6,8 |

|    |                              |  |  |        |
|----|------------------------------|--|--|--------|
| 28 | OMT Class description model  | Модель описания классов                              | Используется для описания внутренней структуры объектно - ориентированного класса.   | 5      |
| 29 | OMT Data value decomposition | OMT Декомпозиция значений данных                     | Содержит большие референтные модели, включающие процессные модели, элементы организационного вида и т. д. Иллюстрирует связь компонент во время процесса.  | 5      |
| 30 | OMT Dynamic model            | OMT Динамическая модель                              | Описывает временные, поведенческие и управляющие аспекты системы, т.е. описывает последовательность операций с помощью последовательности событий.   | 5      |
| 31 | OMT Functional model         | OMT Функциональная модель                            | Описывает функциональные аспекты системы, преобразующие входные значения в выходные.   | 5      |
| 32 | OMT Object model             | OMT Объектная модель                                 | Показывает структуру объектов и их атрибуты. Описывает статичные, структурные, а также ориентированные на данные аспекты системы. Позволяет отображать структуру объектов, их отношения к другим объектам и их атрибуты. | 5      |
| 33 | PCD (Process Chain Diagram)  | Диаграмма цепочек процессов                          | Описывает законченные цепочки процессов, а также их связи с объектами моделей других видов.  | 2,5, 6 |
| 34 | PCD with material flow       | Диаграмма цепочек процессов с материальными потоками | Описывает законченные цепочки процессов, а также их связи с объектами моделей других видов, в том числе с материалами, продуктами и услугами.  | 2,5    |
| 35 | PPC (Project Process Chain)  | Цепочка процесса проектирования                      | Описывает связь ARIS и MS Project. Планирует ресурсы и время в рамках управления проектами.  | 5      |
| 36 | Process instantiation model  | Модель инициации процесса                            | Описывает условия инициации (время начала, продолжительность, периодичность и т.д.) выполнения БП при их имитационном моделировании.   | 2,5    |
| 37 | Process selection diagram    | Диаграмма выбора процесса                            | Показывает различные процессные сценарии, размещения индивидуальных сценариев в главном сценарии.  | 1,5    |
| 38 | Process selection matrix     | Матрица выбора процесса                              | Описывает различные сценарии процессов, привязывая основные процессы к отдельным сценариям.  | 5-8    |
| 39 | Product allocation diagram   | Диаграмма окружения продукта                         | Описывает, какие ОЕ используют или производят те или иные виды продукции, а также, какие функции требуются для создания продукции или для каких функций продукция является входной.                                      | 5      |
| 40 | Product selection matrix     | Матрица выбора продукта                              | Описывает ОЕ и продукцию, за которую ОЕ несет ответственность. Продукты связываются с их создающими функциями.   | 5      |
| 41 | Product tree                 | Дерево продуктов                                     | Описывает номенклатуру продукции.  | 5      |



|    |  |  |   |     |
|----|--|--|---|-----|
| 42 | Product/Service exchange diagram           | Диаграмма движения продуктов/услуг               | Описывает создание продуктов (услуг), а также их обмена (движения) внутри предприятия.  | 5,6 |
| 43 | Product/Service exchange diagram (graphic) | Диаграмма движения продуктов/услуг (графическая) | Описывает то же, что и Диаграмма движения продуктов/услуг, но в пиктографическом виде.  | 5,6 |
| 44 | Product/Service tree                       | Дерево продуктов/услуг                           | Описывает процесс выпуска продукции из составляющих ее частей.  | 5,6 |
| 45 | Product/Service tree (graphic)             | Дерево продуктов/услуг (графическое)             | Описывает то же, что и Дерево продуктов/услуг, но в пиктографическом виде.  | 5,6 |
| 46 | Program flow-chart (PF)                    | Блок-схема программы                             | На этапе спецификации описывает последовательность выполнения процедур в программе.   | 5   |
| 47 | Program structure chart                    | Диаграмма структуры программы                    | На этапе спецификации описывает все взаимодействия типов прикладных систем, типов модулей и типов ИТ-функций при помощи других типов ARIS-диаграмм, с отражением логики взаимодействия.                                   | 5   |
| 48 | RAD  | Диаграмма окружения роли                         | Описывает ролевые аспекты процессов.  | 1,5 |
| 49 | RAMS                                       | Метод RAMS                                       | RAMS (Requirements Analysis for Manufacturing Systems) или в русском переводе АТПС (Анализ Требований для Производственных Систем) – это метод анализа деятельности компании, предложенный фирмой Digital Equipment.      | 5   |
| 50 | Rights diagram                             | Диаграмма прав                                   | Описывает процессы и лиц (групп), уполномоченных их инициировать.   | 5   |
| 51 | Role diagram                               | Диаграмма ролей                                  | Используется для более детального описания процесса. Внимание заостряется на участии ОЕ в процессе и их ролях. Объекты и их взаимосвязи характеризуются с точки зрения ролей участников в процессе и условий авторизации. | 5   |
| 52 | Rule diagram                               | Диаграмма правил                                 | Используется для снижения сложности процессных моделей, а именно для описания сложной бизнес-логики, на которую создается ссылка из процессной модели.  | 5   |
| 53 | SAP ALE filter model                       | SAP ALE Модель фильтров                          | Описывает правила, по которым в приложении отбираются типы функций. Устанавливает связь между функциями и SAP-приложениями.   | 5   |
| 54 | SAP ALE message flow model                 | SAP ALE Модель потоков сообщений                 | Описывает потоки сообщений и их направление между типами прикладных систем и типами функций.  | 5   |

|    |                               |  |   |     |
|----|-------------------------------|--|---|-----|
| 55 | SAP ALE message type model    | SAP ALE Модель типов сообщений                     | В зависимости от передаваемых данных и транзакций классифицирует сообщения, которыми обмениваются распределенные элементы систем.   | 5   |
| 56 | Screen diagram                | Диаграмма экранов                                  | Описывает экранные формы при разработке ПО с целью автоматизации процесса порождения экранных форм из диаграмм экранов.   | 5   |
| 57 | Structuring model             | Структурная модель                                 | Описывает иерархическое или другое системное упорядочения фактов (специализацию или обобщение фактов), где единственный используемый объект (структурный элемент) представляет некоторый факт.                  | 3,5 |
| 58 | UML Activity diagram          | UML диаграмма активности                           | Описывает процессы как последовательности выполнения отдельных операций.  | 5,6 |
| 59 | UML Class description diagram | UML диаграмма описания классов                     | Более полно и точно описывает статические связи между элементами модели.  | 5,6 |
| 60 | UML Class diagram             | UML диаграмма классов                              | Описывает статические связи между элементами модели, такими как класс, объект и интерфейс.  | 5,6 |
| 61 | UML Collaboration diagram     | UML диаграмма взаимодействия                       | Описывает взаимодействия между объектами в форме обмена сообщениями.  | 5,6 |
| 62 | UML Component diagram         | UML диаграмма компонентов                          | Описывает взаимосвязи между программными и аппаратными компонентами системы.  | 5,6 |
| 63 | UML Statechart diagram        | UML диаграмма состояний                            | Описывает иерархическую структуру состояний объектов системы и переходы между состояниями под воздействием определенных событий.  | 5,6 |
| 64 | UML Use case diagram          | UML диаграмма сценариев                            | Описывает внешнее поведение системы с точки зрения пользователя, т. е. описывает варианты использования системы.  | 5,6 |
| 65 | Value-added chain diagram     | Диаграмма цепочки добавленной стоимости (качества) | Описывает последовательность и иерархию функций, непосредственно участвующих в переделах информации или продукции и как следствие в формировании добавленного качества и величины добавленной стоимости выхода. | 2-8 |



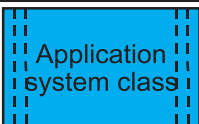
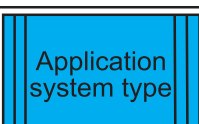

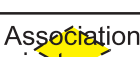
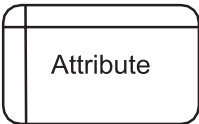
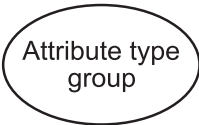

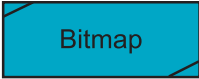
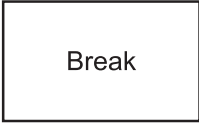

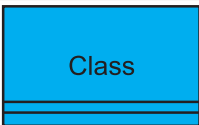
Здесь в последнем столбце перечислены следующие номера фильтров, реализованных в ARIS 5.0:

1. ARIS for MySAP.com;
2. ARIS Simulation and ARIS ABC Filter;
3. Balanced Scorecard Filter;
4. Easy Filter;
5. Entier Method (Fullmethod);
6. Extended Standard Filter;
7. SAP Filter;
8. Standard Filter.






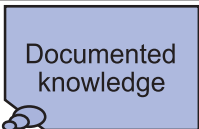

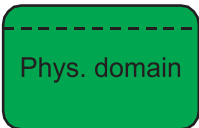

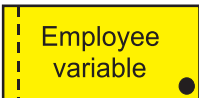
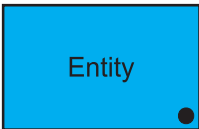
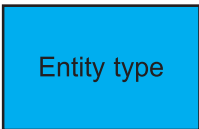
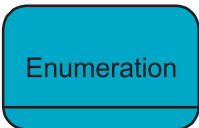


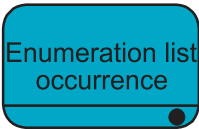

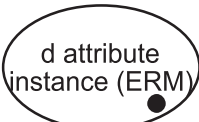
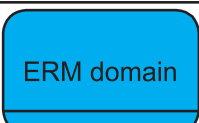
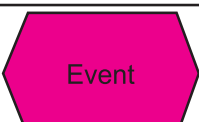
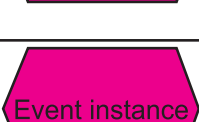
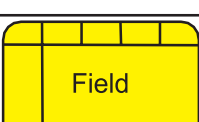
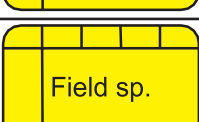
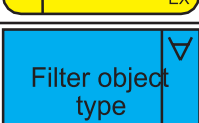


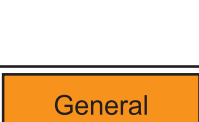
## ПРИЛОЖЕНИЕ 2


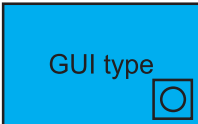



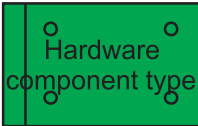

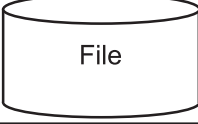

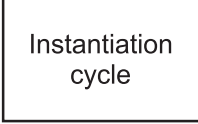
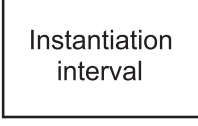
### Краткое описание основных ARIS-объектов

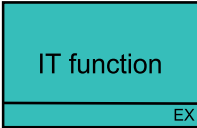
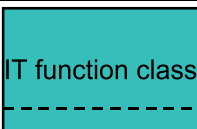
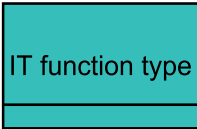

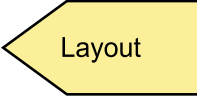
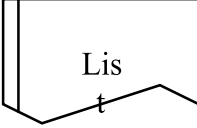

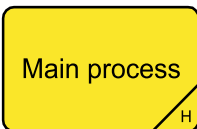
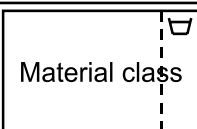
| Тип объекта   | Перевод                  | Краткое описание назначения  |
|---|--------------------------|--|
|  Actor                     | Действующий субъект      | Активизирует потоки данных, создавая либо поглощая данные. Является либо источником, либо потребителем в этом потоке, т. е. связан с его входом или выходом.   |
|  Application system        | Прикладная система       | Конкретизирует понятие "тип прикладной системы" до уровня уникального экземпляра, идентифицируемого, например, по номеру версии (лицензии). Компания может владеть несколькими прикладными системами (версиями), относящимися к одному типу.             |
|  Application system class  | Класс прикладной системы | Объединяет схожие типы прикладных систем. Принцип схожести определяется выбранным критерием классификации. Конкретный тип прикладной системы можно отнести к нескольким классам, используя различные критерии.   |
|  Application system type  | Тип прикладной системы   | Отражает обобщение отдельных прикладных систем, обладающих одинаковыми техническими и функциональными характеристиками.  |
|  Association             | Ассоциация               | Представляет группу связей, одинаковых по структуре и семантике.   |
|  Association instance    | Ассоциация (экземпляр)   | Отображает связи между объектами (экземплярами).   |
|  Attribute               | Атрибут                  | Модель реляционных таблиц использует понятия "атрибут" и "реляционная таблица" для описания логической структуры БД. В этом случае атрибут характеризует свойства реляционной таблицы.   |
|  Attribute type group    | Группа типов атрибутов   | Отображает группу ERM-атрибутов одного типа сущности, тесно связанных между собой по семантике. Это позволяет, например, создать группу, содержащую несколько ERM-атрибутов, образующих вместе вторичный ключ.   |
|  Authorization condition | Полномочие               | Например, в моделях eEPC или PCD накладывает ограничения на выполнение определенных действий организационными единицами.   |
|  Bitmap                  | Точечный рисунок         | Содержит 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            | Сложный объект, параметры которого используются для поддержки различных БП.  |
|  Class                   | Класс                    | Группа объектов со схожими характеристиками и отношениями. Классы отражают основные структуры моделируемого приложения.  |

|  |                         |   |
|--|-------------------------|---|
|  | Критерий классификации  | Определяет критерии, согласно которым следует объединить рассматриваемые объекты.   |
|  | Кластер (экземпляр)     | Экземпляр объекта кластер/модель данных. Он представляет собой логический взгляд на набор объектов данных или структур.   |
|  | Кластер / модель данных | Отражает логический взгляд на набор типов сущностей и типов отношений в модели данных, что требуется для описания сложных объектов.   |
|  | Столбец                 | В модели экранного интерфейса может быть произведено графическое деление поверхности на области. Столбец соотносится с полученными таким образом вертикальными частями.                     |
|  | Взаимодействие          | Является частью диаграммы взаимодействий и определяет тип информации, которой обмениваются две организационные единицы, а также участников этого обмена.                                    |
|  | Сложный объект          | Отображает логически связанную часть структуры модели данных.   |
|  | Тип сложного объекта    | Обобщает несколько схожих сложных объектов.   |
|  | Коннектор               | Отображает разделение потока данных на несколько самостоятельных потоков.   |
|  | Ограничение             | Позволяет налагать ограничения на диапазоны значений для экземпляров атрибутов.   |
|  | Управляющее воздействие | Набор мер, направленных на устранение возможных причин возникновения рисков или на снижение последствий, связанных с появлением риска.  |
|  | Категория затрат        | Отображает группу платежей одного типа, т.е. расходы, обусловленные одинаковыми причинами.  |
|  | Стоимостный фактор      | Единица измерения для результата (выхода) функции, например, количество обработанных заявок, потраченное на работу время и т.д.   |
|  | COT-атрибут             | Описывает атрибуты, которые не могут быть точно определены. Эти атрибуты следует рассматривать в качестве отдельных объектов и определять как текстовые поля ввода на экранных интерфейсах. |
|  | COT-атрибут (экземпляр) | Описывает COT-атрибут в отдельно взятом случае, т. е. ситуация, рассмотренная на более высоком уровне в общем виде, теперь описывается более детально.                                      |

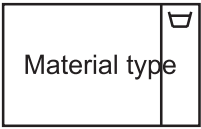


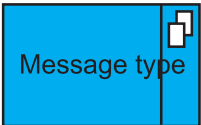
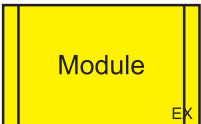
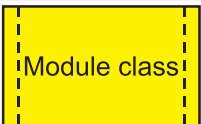
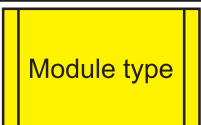

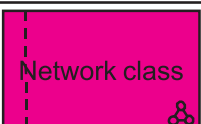

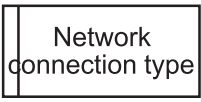



|   |                                  |  |
|---|----------------------------------|--|
|    | <i>Критический фактор</i>        | Определяет те аспекты деятельности компании, которые следует принять во внимание для достижения поставленной цели. Они приписываются целям компании на диаграмме целей.  |
|    | <i>Хранилище данных</i>          | Является средством хранения информации для последующего доступа к ней.   |
|    | <i>Значения данных</i>           | Связывают выход одного БП (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). |
|    | <i>СУБД</i>                      | Система управления базами данных.  |
|   | <i>Тип СУБД</i>                  | Отображает критерий классификации СУБД с одинаковыми техническими характеристиками. Как правило, типы СУБД идентифицируются по имени и номеру версии.  |
|  | <i>Документированное знание</i>  | Позволяет специфицировать знания, содержащиеся в документах, чертежах и относящиеся к заданной категории знаний.   |
|  | <i>Диапазон</i>                  | Позволяет отобразить диапазон значений атрибута на модели реляционных таблиц, т. е. множество принимаемых им значений.   |
|  | <i>Диапазон (физический)</i>     | Отражает диапазон значений содержимого ячейки таблицы на диаграмме таблиц, т.е. множество принимаемых этой ячейкой значений.   |
|  | <i>Эскиз списка</i>              | 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.  |
|  | <i>Зарезервированная позиция</i> | Зарезервированное в БП место для сотрудника, который будет специфицирован позже.   |
|  | <i>Сущность</i>                  | Реальный или абстрактный объект, представляющий интерес в рамках рассматриваемых целей и задач компании.   |
|  | <i>Тип сущности</i>              | Объединяет сходные сущности. Сущности относятся к одному типу, если они могут быть описаны одними и теми же типовыми характеристиками (атрибутами).  |
|  | <i>Перечисление</i>              | Используется для конкретизации ERM-атрибутов путем перечисления их возможных типовых значений.   |

|   |                             |   |
|---|-----------------------------|---|
|    | Перечисление<br>(экземпляр) | Одно из значений объекта "Перечисление".  |
|    | ERM-атрибут                 | Характеристика, описывающая тип сущности или тип отношения.   |
|    | ERM-атрибут<br>(экземпляр)  | Предназначен для описания сущности или отношения.   |
|    | ERM-диапазон                | Диапазон значений, принимаемых ERM-атрибутом.   |
|    | Событие                     | Состояние, которое является существенным для управления БП и которое оказывает влияние или контролирует дальнейшее развитие одного или более БП. Изменения состояния отражаются с помощью информационных объектов.  |
|   | Экземпляр<br>события        | Конкретное событие, произошедшее при выполнении конкретного БП. 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).  |
|  | Поле                        | Логическая структура СУБД может быть описана с помощью таблиц и полей. Поля определяют логическое содержание таблицы. Значения поля могут быть описаны объектом "Диапазон (физический)".  |
|  | Поле<br>(экземпляр)         | Отображает конкретный экземпляр поля таблицы. Таблицы и поля формируют логическую структуру СУБД.   |
|  | Тип сортировки              | Устанавливает критерии частоты (повторяемости) функций модели, выполняемых системой.  |
|  | Функция                     | Действие или набор действий, выполняемых над исходным объектом (документом, материалом и т. п.) с целью получения заданного результата (документа, материала и т. п.).  |
|  | Экземпляр<br>функции        | 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 первый квартал 2001 года", который является экземпляром для функции "Создание плана расходов на квартал". |
|  | Общий ресурс                | Является необходимым элементом для выполнения функций, но не может быть отнесен к категории трудовых или операционных ресурсов.   |


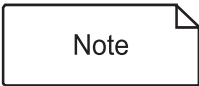
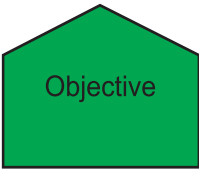
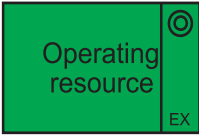
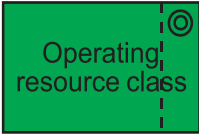
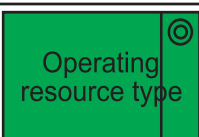
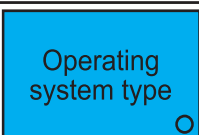





|   |  |   |
|---|--|---|
|  Generalization            | Тип обобщения                            | В процессе моделирования данных устанавливается сходство между объектами (сущностями). Эти объекты затем комбинируются, образуя тип объекта более высокого уровня обобщения.  |
|  GUI type                  | Тип графического интерфейса пользователя | Отображает обобщение индивидуальных интерфейсов пользователя, наделенных одинаковыми характеристиками. Он, как правило, идентифицируется по имени и номеру версии. На основе специфических типов таких интерфейсов реализуются типы прикладных систем.  |
|  Group                     | Группа                                   | Отображает группу сотрудников, работающих вместе в течение определенного промежутка времени, например, проектная группа.  |
|  Hardware component        | Компонент аппаратного обеспечения        | Экземпляр определенного типа оборудования, имеющегося в компании. Он часто идентифицируется при помощи инвентарного номера.   |
|  Hardware component class  | Класс компонента аппаратного обеспечения | Объединяет схожие типы компонентов аппаратного обеспечения. Принцип схожести определяется выбранным критерием классификации. Конкретный тип компонента аппаратного обеспечения можно отнести к нескольким классам, используя различные критерии.  |
|  Hardware component type | Тип компонента аппаратного обеспечения   | 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.   |
|  File                    | 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                           | Принадлежит модели инициации БП и является объектом второго (промежуточного) уровня, служащим для описания повторяющихся последовательностей интервалов инициации.  |
|  Instantiation interval  | Интервал инициации                       | Принадлежит модели инициации БП 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) БП, распределение, циклическая повторяемость, период. Пример использования этого объекта: "Для определенного процесса в 8:00 в рабочий день запустить 50 инициации (реализаций) с равномерным распределением". |

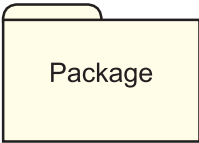
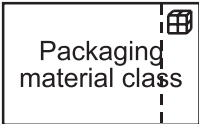



|   |                  |  |
|---|------------------|--|
|    | План инициации   | Принадлежит модели инициации БП и является объектом верхнего уровня, служащим для описания нескольких циклов инициации. Примером такого объекта является "Рабочий план утренней смены".  |
|    | Функция ИТ       | Является 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.  |
|    | Класс функции ИТ | Объединяет схожие типы функций ИТ. Принцип схожести определяется выбранным критерием классификации. Конкретный тип функции ИТ можно отнести к нескольким классам, используя различные критерии.  |
|    | Тип функции ИТ   | Если рассматривать транзакции, то тип функции ИТ 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 ИТ отражает обобщение отдельных функций ИТ, обладающих сходными техническими характеристиками. |
|   | Категория знаний | Используется для классификации знаний по тематикам и областям использования.   |
|  | Схема оформления | Характеристики графического оформления отдельных элементов экранного интерфейса. Характеризуется цветом фона, стилем отображения текста и т. д.  |
|  | 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.                 |
|  | Расположение     | Определяет физическое местонахождение организационных единиц, оборудования и технических ресурсов компании. Им может быть регион, город, завод, здание, комната и даже отдельное рабочее место.  |
|  | Начало цикла     | Располагается на блок-схеме программы и определяет начало последовательности, состоящей из нескольких шагов. Последовательность завершается окончанием цикла.  |
|  | 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").   |
|  | 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.   |
|  | Поток материалов | Объект, отражающий движение материалов от одной функции (процесса) к другой функции (процессу).  |

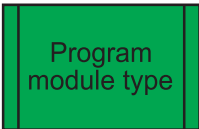

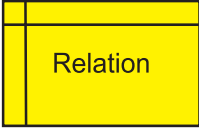


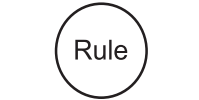
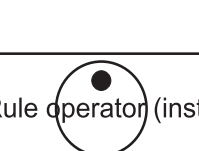
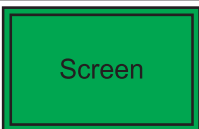
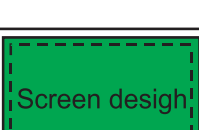
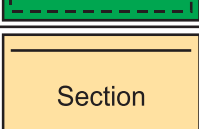

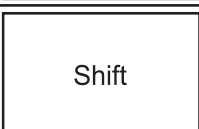


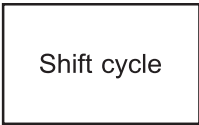
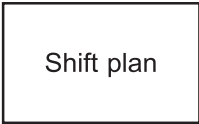



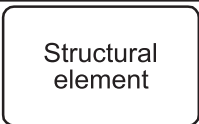

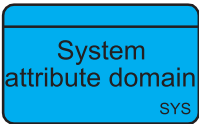


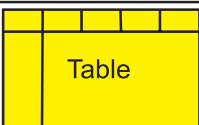
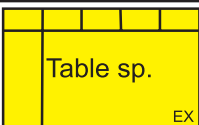

|   |                         |  |
|---|-------------------------|--|
|  Material type             | Тип материала           | Тип материала отображает обобщение материалов с одинаковыми характеристиками.  |
|  Memory location           | Ячейка памяти           | Предназначены для хранения данных в виде таблиц или полей.   |
|  Message flow              | Поток сообщений         | Используется для более детального описания связей, по которым объекты обмениваются сообщениями.  |
|  Message type              | Тип сообщения           | Классифицирует потоки между системами в соответствии с передаваемыми данными и процессами, в которых они задействованы.  |
|  Module                    | 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            | Объединяет схожие типы модулей. Принцип схожести определяется выбранным критерием классификации. Конкретный тип модуля можно отнести к нескольким классам, используя различные критерии.           |
|  Module type             | Тип модуля              | Представляет собой независимо исполняемую часть типа прикладной системы и отражает обобщение отдельных модулей, имеющих одинаковые технические характеристики.                                     |
|  Network                 | Сеть                    | Представляет собой экземпляр, относящийся к одному из типов сетей, имеющихся в компании. Сеть может быть идентифицирована по своему расположению либо по инвентарному номеру.                      |
|  Network class           | Класс сети              | Схожие типы сетей. Принцип схожести определяется выбранным критерием классификации. Конкретный тип сети можно отнести к нескольким классам, используя различные критерии.                          |
|  Network connection      | 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 | Представляет собой обобщение отдельных сетевых соединений с одинаковыми техническими характеристиками.   |
|  Network node            | 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           | Представляет собой обобщение отдельных узлов сети с одинаковыми техническими характеристиками.   |
|  Network protocol        | Сетевой протокол        | Стандартизованный метод связи и обмена информацией между прикладными системами в сети. Для каждого типа сети должен быть определен сетевой протокол.   |

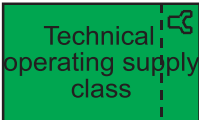
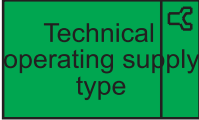
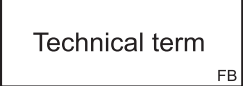
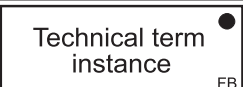
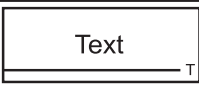

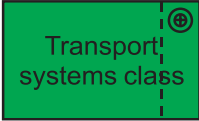
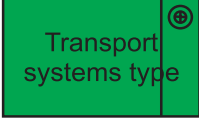








|   |                             |   |
|---|-----------------------------|---|
|    | Тип сети                    | Отражает обобщение отдельных (информационных) сетей с одинаковыми техническими характеристиками.  |
|    | Примечание                  | Принадлежит UML-моделям и позволяет связывать с объектами дополнительную информацию.  |
|    | Цель                        | Определяет цели организации, которые (предположительно) будут достигнуты при наличии соответствующих факторов успеха и реализации новых БП.   |
|    | 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).   |
|    | Класс операционного ресурса | Объединяет схожие типы операционных ресурсов. Принцип схожести определяется выбранным критерием классификации. Конкретный тип операционного ресурса можно отнести к нескольким классам, используя различные критерии.   |
|  | Тип операционного ресурса   | Отражает обобщение отдельных операционных ресурсов, обладающих одинаковыми техническими характеристиками.   |
|  | Тип операционной системы    | Отражает обобщение отдельных операционных систем, обладающих одинаковыми техническими характеристиками. Операционные системы, как правило, идентифицируются по имени и номеру версии.   |
|  | Операция                    | Отражают функциональность или трансформации данных (атрибутов), связанные с классом.  |
|  | Организационная схема       | Отражает совокупность организационных взаимосвязей, рассматриваемых на верхнем уровне абстракции.   |
|  | 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. |
|  | Организационная единица     | Являются исполнителями задач, решение которых необходимо для достижения бизнес-целей. Это достаточно стабильные образования, представленные набором штатных единиц, занимаемых конкретными сотрудниками компании.   |
|  | Тип организационной единицы | Обобщает отдельные организационные единицы, обладающие 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.  |

|   |                                     |  |
|---|-------------------------------------|--|
|    | <i>Пакет</i>                        | 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. |
|    | <i>Класс упаковочного материала</i> | Объединяет типы упаковочных материалов. При таком объединении используются различные критерии классификации. Один тип упаковочного материала может относиться к нескольким классам материалов.   |
|    | <i>Тип упаковочного материала</i>   | Совокупность отдельных упаковочных материалов, имеющих одинаковые характеристики.  |
|    | <i>Страница</i>                     | Делит экранный интерфейс на страницы.  |
|   | <i>Параметр</i>                     | Отображает входные и выходные параметры функции IT.  |
|  | <i>Сотрудник</i>                    | Отдельный служащий компании, идентифицируемый, например, по его персональному коду. Он может быть связан с организационными единицами (в которые он входит), а также с функциями (которые он исполняет или за которые отвечает).   |
|  | <i>Тип сотрудника</i>               | 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.  |
|  | <i>Должность</i>                    | Элементарная организационная единица компании. С ней связаны сотрудники, и, как правило, их права и обязанности определяются именно профилем должности.  |
|  | <i>Процесс</i>                      | Отображает функцию, преобразующую значения данных. Результат процесса зависит от поведения всей системы в целом.   |
|  | <i>Продукт/услуга</i>               | 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.  |
|  | <i>Библиотека приложений</i>        | Содержит программы и/или программные модули, требуемые для выполнения задачи.  |
|  | <i>Программный модуль</i>           | Отдельный программный файл, расположенный на носителе данных (например, файл EXE или COM) и приобретаемый вместе с лицензией.  |

|   |                              |  |
|---|------------------------------|--|
|    | Тип программного модуля      | Отражает обобщение отдельных программных модулей, обладающих одинаковыми техническими характеристиками.  |
|    | Язык программирования        | Отражает обобщение отдельных программных модулей, обладающих одинаковыми техническими характеристиками.  |
|    | Реляционная таблица          | Описывает тип сущности через ее атрибуты. Это подмножество всевозможных комбинаций диапазонов значений атрибутов.  |
|    | Отношение                    | Логическая связь между сущностями. Схожие отношения могут быть объединены, образуя тип отношения.  |
|    | Тип отношения                | Объединяет схожие отношения. Отношения относятся к одному типу, если их можно описать с помощью одинаковых атрибутов.  |
|   | Риск                         | Означает потенциальную опасность для БП не достигнуть желаемой цели.   |
|  | Правило                      | Представляет собой правило разветвления и слияния веток БП. Если перейти к рассмотрению каждой отдельной функции БП, то можно сказать, что правило отражает логическое соотношение между несколькими исходными для функции событиями и несколькими результирующими.  |
|  | Правило (экземпляр)          | Представляет собой логический оператор уровня экземпляров и определяет связь между функцией и событием на этом уровне.   |
|  | Экранный интерфейс           | 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.   |
|  | Таблица экранного интерфейса | Перечисляет объекты и типы объектов на экранном интерфейсе.  |
|  | Раздел                       | Видимая область на модели экранного интерфейса может быть разделена на несколько частей. Данный объект описывает полученные таким образом ряды.  |
|  | Разделитель                  | Пустой ряд или столбец на экранном интерфейсе.   |
|  | Смена                        | Объект модели "Календарь смен", определяющий интервал времени, в течение 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. |

|   |  |  |
|---|--|--|
|    | <i>Цикл смен</i>                             | 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.                                     |
|    | <i>План смен</i>                             | 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. |
|    | <i>Решение</i>                               | Продуманный комплекс управляющих воздействий, обеспечивающий полную нейтрализацию одного или нескольких рисков.  |
|    | <i>Оператор обобщения/специализации</i>      | Оператор обобщения/специализации разделяет или комбинирует схожие классы.  |
|    | <i>Состояние</i>                             | Некоторый набор значений, описывающих реакцию объекта на совокупность событий (воздействий).   |
|   | <i>Структурный элемент</i>                   | 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.  |
|  | <i>System attribute – Системный атрибут</i>  | Описывает объекты, связанные с задачей экспорта данных. Возможно отображение первичных и внешних ключей, обязательных и описательных полей.  |
|  | <i>Диапазон системного атрибута</i>          | Описывает область допустимых значений для системных атрибутов.   |
|  | <i>Системная организационная единица</i>     | Интегрированные прикладные системы содержат организационные структуры, которые следует принять во внимание при внедрении таких систем. Данный объект используется при моделировании этих организационных структур.                               |
|  | <i>Тип системной организационной единицы</i> | Обобщает отдельные системные организационные единицы с похожими характеристиками. Схожесть может, например, заключаться в правах доступа к прикладной системе.   |
|  | <i>Таблица</i>                               | Логическая структура СУБД может быть описана с помощью таблиц и полей. Таблица представляет собой обобщение отдельных экземпляров таблиц, имеющих одинаковую логическую структуру.   |
|  | <i>Таблица (экземпляр)</i>                   | 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.       |
|  | <i>Вспомогательные технические средства</i>  | Являются экземпляром определенного типа вспомогательных технических средств.   |

|   |  |   |
|---|--|---|
|    | Класс вспомога-<br>тельных<br>технических<br>средств | Объединение на основе определенного критерия классификации схожих типов вспомогательных технических средств.  |
|    | Тип вспомога-<br>тельных<br>технических<br>средств   | Обобщает отдельные экземпляры вспомогательных технических средств, обладающих одинаковыми техническими характеристиками.  |
|    | Технический<br>термин                                | Отражает концептуальный взгляд на имеющиеся в организации информационные объекты и используется для выделения специфичных терминов и понятий, а также существующих между ними взаимосвязей.   |
|    | Технический<br>термин<br>(экземпляр)                 | Представляет собой технический термин на уровне экземпляров.  |
|    | Текст  | Содержит общее текстовое описание, включенное в соответствующую модель.   |
|    | Транспортная<br>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. |
|  | Класс<br>транспортной<br>системы                     | Объединение схожих типов транспортных систем. Объединение происходит на основе выбранного критерия классификации.   |
|  | Тип<br>транспортной<br>системы                       | Отражает обобщение отдельных экземпляров транспортных систем, обладающих одинаковыми техническими характеристиками.   |
|  | Пример<br>использования                              | Присутствует в моделях UML и описывает случаи использования, при которых действующий субъект совершает определенные действия.   |
|  | Взгляд   | Отражает логический взгляд на набор реляционных таблиц и атрибутов. К примеру, представление менеджера по продажам о понятии "Заказ" может отличаться от видения кладовщика.  |
|  | Взгляд<br>(физический)                               | Отражает логический взгляд на набор таблиц и полей.   |
|  | Складское<br>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.  |
|  | Класс<br>складского<br>оборудования                  | Отражает обобщение отдельных экземпляров складского оборудования, обладающих одинаковыми техническими характеристиками.   |
|  | Тип складского<br>оборудования                       | Объединяет схожее складское оборудование. Объединение происходит на основе выбранного критерия классификации.   |



## ПРИЛОЖЕНИЕ 3

**ТАБЛИЦА ТИПОВ ОТНОШЕНИЙ**

| <i>Тип модели</i> | <i>Тип символа<br/>объекта-<br/>источника<br/>связи</i> | <i>Тип символа<br/>объекта-<br/>приемника<br/>связи</i> | <i>Тип<br/>отношения</i> |
|-------------------|---|---|--------------------------|
| eEPC              | Function  | Function  | is predecessor of        |
| eEPC              | Function  | Event   | creates                  |
| eEPC              | Function  | AND operator  | leads to                 |
| eEPC              | Function  | OR operator   | leads to                 |
| eEPC              | Function  | XOR operator  | leads to                 |
| eEPC              | Function  | Process interface                                       | is predecessor of        |
| eEPC              | Function  | Technical term  | has output of            |
| eEPC              | Function  | Location  | is executed at           |
| eEPC              | Function  | File  | creates output to        |
| eEPC              | Function  | Document  | creates output to        |
| eEPC              | Function  | Card file   | creates output to        |
| eEPC              | Function  | Expertise   | creates output to        |
| eEPC              | Function  | Folder  | creates output to        |
| eEPC              | Function  | Telephone   | creates output to        |
| eEPC              | Function  | Fax   | creates output to        |
| eEPC              | Function  | Bar code  | creates output to        |
| eEPC              | Function  | Magnetic tape   | creates output to        |
| eEPC              | Function  | Microfiche  | creates output to        |
| eEPC              | Function  | Cluster   | has output of            |
| eEPC              | Event   | Function  | activates                |
| eEPC              | Event   | AND operator  | is evaluated by          |
| eEPC              | Event   | OR operator   | is evaluated by          |
| eEPC              | Event   | XOR operator  | is evaluated by          |
| eEPC              | Event   | Process interface                                       | activates                |
| eEPC              | AND operator  | Function  | activates                |
| eEPC              | AND operator  | Event   | leads to                 |
| eEPC              | AND operator  | AND operator  | links                    |
| eEPC              | AND operator  | OR operator   | links                    |
| eEPC              | AND operator  | XOR operator  | links                    |
| eEPC              | AND operator  | Process interface                                       | activates                |
| eEPC              | OR operator   | Function  | activates                |
| eEPC              | OR operator   | Event   | leads to                 |
| eEPC              | OR operator   | AND operator  | links                    |
| eEPC              | OR operator   | OR operator   | links                    |
| eEPC              | OR operator   | XOR operator  | links                    |
| eEPC              | OR operator   | Process interface                                       | activates                |
| eEPC              | XOR operator  | Function  | activates                |
| eEPC              | XOR operator  | Event   | leads to                 |
| eEPC              | XOR operator  | AND operator  | links                    |
| eEPC              | XOR operator  | OR operator   | links                    |

|      |                     |                         |                   |
|------|---------------------|-------------------------|-------------------|
| eEPC | XOR operator        | XOR operator            | links             |
| eEPC | XOR operator        | Process interface       | activates         |
| eEPC | Process interface   | Function                | is predecessor of |
| eEPC | Process interface   | Event                   | creates           |
| eEPC | Process interface   | AND operator            | leads to          |
| eEPC | Process interface   | OR operator             | leads to          |
| eEPC | Process interface   | XOR operator            | leads to          |
| eEPC | Process interface   | Process interface       | is predecessor of |
| eEPC | Process interface   | Technical term          | has output of     |
| eEPC | Process interface   | File                    | creates output to |
| eEPC | Process interface   | Document                | creates output to |
| eEPC | Process interface   | Card file               | creates output to |
| eEPC | Process interface   | Expertise               | creates output to |
| eEPC | Process interface   | Folder                  | creates output to |
| eEPC | Process interface   | Telephone               | creates output to |
| eEPC | Process interface   | Fax                     | creates output to |
| eEPC | Process interface   | Bar code                | creates output to |
| eEPC | Process interface   | Magnetic tape           | creates output to |
| eEPC | Process interface   | Microfiche              | creates output to |
| eEPC | Process interface   | Cluster                 | has output of     |
| eEPC | Technical term      | Function                | is input for      |
| eEPC | Technical term      | Event                   | has state         |
| eEPC | Technical term      | Process interface       | is input for      |
| eEPC | Technical term      | File                    | lies on           |
| eEPC | Technical term      | Document                | lies on           |
| eEPC | Technical term      | Card file               | lies on           |
| eEPC | Technical term      | Expertise               | lies on           |
| eEPC | Technical term      | Folder                  | lies on           |
| eEPC | Technical term      | Telephone               | lies on           |
| eEPC | Technical term      | Fax                     | lies on           |
| eEPC | Technical term      | Bar code                | lies on           |
| eEPC | Technical term      | Magnetic tape           | lies on           |
| eEPC | Technical term      | Microfiche              | lies on           |
| eEPC | Organizational unit | Function                | executes          |
| eEPC | Organizational unit | Process interface       | executes          |
| eEPC | Organizational unit | Technical term          | accesses          |
| eEPC | Organizational unit | Application system type | can be user       |
| eEPC | Organizational unit | Cluster                 | accesses          |
| eEPC | Group               | Function                | executes          |
| eEPC | Group               | Process interface       | executes          |
| eEPC | Group               | Technical term          | accesses          |



|      |                         |                         |                    |
|------|-------------------------|-------------------------|--------------------|
| eEPC | Group                   | Application system type | can be user        |
| eEPC | Group                   | Cluster                 | accesses           |
| eEPC | Position                | Function                | executes           |
| eEPC | Position                | Process interface       | executes           |
| eEPC | Position                | Technical term          | accesses           |
| eEPC | Position                | Application system type | can be user        |
| eEPC | Position                | Cluster                 | accesses           |
| eEPC | File                    | Function                | provides input for |
| eEPC | File                    | Event                   | has state          |
| eEPC | File                    | Process interface       | provides input for |
| eEPC | Document                | Function                | provides input for |
| eEPC | Document                | Event                   | has state          |
| eEPC | Document                | Process interface       | provides input for |
| eEPC | Card file               | Function                | provides input for |
| eEPC | Card file               | Event                   | has state          |
| eEPC | Card file               | Process interface       | provides input for |
| eEPC | Expertise               | Function                | provides input for |
| eEPC | Expertise               | Event                   | has state          |
| eEPC | Expertise               | Process interface       | provides input for |
| eEPC | Folder                  | Function                | provides input for |
| eEPC | Folder                  | Event                   | has state          |
| eEPC | Folder                  | Process interface       | provides input for |
| eEPC | Telephone               | Function                | provides input for |
| eEPC | Telephone               | Event                   | has state          |
| eEPC | Telephone               | Process interface       | provides input for |
| eEPC | Fax                     | Function                | provides input for |
| eEPC | Fax                     | Event                   | has state          |
| eEPC | Fax                     | Process interface       | provides input for |
| eEPC | Bar code                | Function                | provides input for |
| eEPC | Bar code                | Event                   | has state          |
| eEPC | Bar code                | Process interface       | provides input for |
| eEPC | Magnetic tape           | Function                | provides input for |
| eEPC | Magnetic tape           | Event                   | has state          |
| eEPC | Magnetic tape           | Process interface       | provides input for |
| eEPC | Microfiche              | Function                | provides input for |
| eEPC | Microfiche              | Event                   | has state          |
| eEPC | Microfiche              | Process interface       | provides input for |
| eEPC | Application system type | Function                | can support        |
| eEPC | Application system type | Process interface       | can support        |
| eEPC | Application system type | Technical term          | can use            |
| eEPC | Application system type | Cluster                 | can use            |
| eEPC | Cluster                 | Function                | is input for       |

|                             |                     |                         |                   |
|-----------------------------|---------------------|-------------------------|-------------------|
| eEPC                        | Cluster             | Event                   | has state         |
| eEPC                        | Cluster             | Process interface       | is input for      |
| eEPC                        | Cluster             | File                    | lies on           |
| eEPC                        | Cluster             | Document                | lies on           |
| eEPC                        | Cluster             | Card file               | lies on           |
| eEPC                        | Cluster             | Expertise               | lies on           |
| eEPC                        | Cluster             | Folder                  | lies on           |
| eEPC                        | Cluster             | Telephone               | lies on           |
| eEPC                        | Cluster             | Fax                     | lies on           |
| eEPC                        | Cluster             | Bar code                | lies on           |
| eEPC                        | Cluster             | Magnetic tape           | lies on           |
| eEPC                        | Cluster             | Microfiche              | lies on           |
| eEPC                        | Internal person     | Function                | executes          |
| eEPC                        | Internal person     | Process interface       | executes          |
| eEPC                        | Internal person     | Technical term          | accesses          |
| eEPC                        | Internal person     | Application system type | can be user type  |
| eEPC                        | Internal person     | Cluster                 | accesses          |
| eEPC                        | External person     | Function                | executes          |
| eEPC                        | External person     | Process interface       | executes          |
| eEPC                        | External person     | Technical term          | accesses          |
| eEPC                        | External person     | Application system type | can be user type  |
| eEPC                        | External person     | Cluster                 | accesses          |
| eEPC                        | Person type         | Function                | executes          |
| eEPC                        | Person type         | Process interface       | executes          |
| eEPC                        | Person type         | Technical term          | can access        |
| eEPC                        | Person type         | Application system type | can be user type  |
| eEPC                        | Person type         | Cluster                 | can access        |
| Function allocation diagram | Function            | Location                | is executed at    |
| Function allocation diagram | Function            | Technical term          | has output of     |
| Function allocation diagram | Function            | Card file               | creates output to |
| Function allocation diagram | Function            | File                    | creates output to |
| Function allocation diagram | Function            | Document                | creates output to |
| Function allocation diagram | Function            | Expertise               | creates output to |
| Function allocation diagram | Function            | Folder                  | creates output to |
| Function allocation diagram | Function            | Bar code                | creates output to |
| Function allocation diagram | Function            | Microfiche              | creates output to |
| Function allocation diagram | Function            | Telephone               | creates output to |
| Function allocation diagram | Function            | Fax                     | creates output to |
| Function allocation diagram | Function            | Magnetic tape           | creates output to |
| Function allocation diagram | Function            | Cluster                 | has output of     |
| Function allocation diagram | Organizational unit | Function                | executes          |
| Function allocation diagram | Organizational unit | Technical term          | accesses          |

|                             |                         |                         |                    |
|-----------------------------|-------------------------|-------------------------|--------------------|
| Function allocation diagram | Organizational unit     | Application system type | can be user        |
| Function allocation diagram | Organizational unit     | Cluster                 | accesses           |
| Function allocation diagram | Position                | Function                | executes           |
| Function allocation diagram | Position                | Technical term          | accesses           |
| Function allocation diagram | Position                | Application system type | can be user        |
| Function allocation diagram | Position                | Cluster                 | accesses           |
| Function allocation diagram | Internal person         | Function                | executes           |
| Function allocation diagram | Internal person         | Technical term          | accesses           |
| Function allocation diagram | Internal person         | Application system type | can be user        |
| Function allocation diagram | Internal person         | Cluster                 | accesses           |
| Function allocation diagram | External person         | Function                | executes           |
| Function allocation diagram | External person         | Technical term          | accesses           |
| Function allocation diagram | External person         | Application system type | can be user        |
| Function allocation diagram | External person         | Cluster                 | accesses           |
| Function allocation diagram | Group                   | Function                | executes           |
| Function allocation diagram | Group                   | Technical term          | accesses           |
| Function allocation diagram | Group                   | Application system type | can be user        |
| Function allocation diagram | Group                   | Cluster                 | accesses           |
| Function allocation diagram | Technical term          | Function                | is input for       |
| Function allocation diagram | Technical term          | Card file               | lies on            |
| Function allocation diagram | Technical term          | File                    | lies on            |
| Function allocation diagram | Technical term          | Document                | lies on            |
| Function allocation diagram | Technical term          | Expertise               | lies on            |
| Function allocation diagram | Technical term          | Folder                  | lies on            |
| Function allocation diagram | Technical term          | Bar code                | lies on            |
| Function allocation diagram | Technical term          | Microfiche              | lies on            |
| Function allocation diagram | Technical term          | Telephone               | lies on            |
| Function allocation diagram | Technical term          | Fax                     | lies on            |
| Function allocation diagram | Technical term          | Magnetic tape           | lies on            |
| Function allocation diagram | Card file               | Function                | provides input for |
| Function allocation diagram | File                    | Function                | provides input for |
| Function allocation diagram | Document                | Function                | provides input for |
| Function allocation diagram | Expertise               | Function                | provides input for |
| Function allocation diagram | Folder                  | Function                | provides input for |
| Function allocation diagram | Bar code                | Function                | provides input for |
| Function allocation diagram | Microfiche              | Function                | provides input for |
| Function allocation diagram | Telephone               | Function                | provides input for |
| Function allocation diagram | Fax                     | Function                | provides input for |
| Function allocation diagram | Magnetic tape           | Function                | provides input for |
| Function allocation diagram | Application system type | Function                | can support        |

|                             |                          |                          |                              |
|-----------------------------|--------------------------|--------------------------|------------------------------|
| Function allocation diagram | Application system type  | Technical term           | can use                      |
| Function allocation diagram | Application system type  | Cluster                  | can use                      |
| Function allocation diagram | Cluster                  | Function                 | is input for                 |
| Function allocation diagram | Cluster                  | Card file                | lies on                      |
| Function allocation diagram | Cluster                  | File                     | lies on                      |
| Function allocation diagram | Cluster                  | Document                 | lies on                      |
| Function allocation diagram | Cluster                  | Expertise                | lies on                      |
| Function allocation diagram | Cluster                  | Folder                   | lies on                      |
| Function allocation diagram | Cluster                  | Bar code                 | lies on                      |
| Function allocation diagram | Cluster                  | Microfiche               | lies on                      |
| Function allocation diagram | Cluster                  | Telephone                | lies on                      |
| Function allocation diagram | Cluster                  | Fax                      | lies on                      |
| Function allocation diagram | Cluster                  | Magnetic tape            | lies on                      |
| Function allocation diagram | Person type              | Function                 | executes                     |
| Function allocation diagram | Person type              | Technical term           | can access                   |
| Function allocation diagram | Person type              | Application system type  | can be user                  |
| Function allocation diagram | Person type              | Cluster                  | can access                   |
| Function tree               | Function                 | Function                 | is process-oriented superior |
| Industrial process          | Function (Manufacturing) | Function (Manufacturing) | is predecessor of            |
| Industrial process          | Function (Manufacturing) | Event                    | creates                      |
| Industrial process          | Function (Manufacturing) | Rule                     | leads to                     |
| Industrial process          | Function (Manufacturing) | Printer                  | creates output to            |
| Industrial process          | Function (Manufacturing) | Document                 | creates output to            |
| Industrial process          | Function (Manufacturing) | Diskette                 | creates output to            |
| Industrial process          | Function (Manufacturing) | File cabinet             | creates output to            |
| Industrial process          | Function (Manufacturing) | Notepad                  | creates output to            |
| Industrial process          | Function (Manufacturing) | Telephone                | creates output to            |
| Industrial process          | Function (Manufacturing) | E-mail                   | creates output to            |
| Industrial process          | Function (Manufacturing) | File bin                 | creates output to            |
| Industrial process          | Function (Manufacturing) | Location                 | is executed at               |
| Industrial process          | Function (Manufacturing) | Packaging Material       | requires                     |

|                    |                          |                          |                    |
|--------------------|--------------------------|--------------------------|--------------------|
| Industrial process | Function (Manufacturing) | Airplane                 | accesses           |
| Industrial process | Function (Manufacturing) | Truck                    | accesses           |
| Industrial process | Function (Manufacturing) | Transport system         | accesses           |
| Industrial process | Function (Manufacturing) | Warehouse equipment      | requires           |
| Industrial process | Function (Manufacturing) | Tool                     | requires           |
| Industrial process | Event                    | Function (Manufacturing) | activates          |
| Industrial process | Event                    | Rule                     | is evaluated by    |
| Industrial process | Rule                     | Function (Manufacturing) | activates          |
| Industrial process | Rule                     | Event                    | leads to           |
| Industrial process | Rule                     | Rule                     | links              |
| Industrial process | Computer                 | Function (Manufacturing) | can support        |
| Industrial process | Printer                  | Function (Manufacturing) | provides input for |
| Industrial process | Document                 | Function (Manufacturing) | provides input for |
| Industrial process | Diskette                 | Function (Manufacturing) | provides input for |
| Industrial process | File cabinet             | Function (Manufacturing) | provides input for |
| Industrial process | Notepad                  | Function (Manufacturing) | provides input for |
| Industrial process | Telephone                | Function (Manufacturing) | provides input for |
| Industrial process | E-mail                   | Function (Manufacturing) | provides input for |
| Industrial process | File bin                 | Function (Manufacturing) | provides input for |
| Industrial process | Organizational unit      | Function (Manufacturing) | executes           |
| Industrial process | Organizational unit      | Computer                 | can be user        |
| Industrial process | Person type              | Function (Manufacturing) | executes           |
| Industrial process | Person type              | Computer                 | can be user        |
| Industrial process | Person (m)               | Function (Manufacturing) | executes           |
| Industrial process | Person (m)               | Computer                 | can be user        |
| Industrial process | Person (f)               | Function (Manufacturing) | executes           |
| Industrial process | Person (f)               | Computer                 | can be user        |

|                    |                     |                          |                          |
|--------------------|---------------------|--------------------------|--------------------------|
| Industrial process | Group               | Function (Manufacturing) | executes                 |
| Industrial process | Group               | Computer                 | can be user              |
| Industrial process | Position            | Function (Manufacturing) | executes                 |
| Industrial process | Position            | Computer                 | can be user              |
| Industrial process | Airplane            | Computer                 | can subsume              |
| Industrial process | Truck               | Computer                 | can subsume              |
| Industrial process | Transport system    | Computer                 | can subsume              |
| Industrial process | Machine             | Function (Manufacturing) | is operating resource of |
| Industrial process | Machine             | Computer                 | can subsume              |
| Industrial process | Robot               | Function (Manufacturing) | is operating resource of |
| Industrial process | Robot               | Computer                 | can subsume              |
| Industrial process | Warehouse equipment | Computer                 | can subsume              |
| Industrial process | Tool                | Computer                 | can subsume              |
| Industrial process | Material type       | Function (Manufacturing) | is partly consumed       |
| Industrial process | Material type       | Function (Manufacturing) | is partly consumed       |
| Office process     | Event               | Function (Office)        | activates                |
| Office process     | Event               | Rule                     | is evaluated by          |
| Office process     | Function (Office)   | Event                    | creates                  |
| Office process     | Function (Office)   | Function (Office)        | is predecessor of        |
| Office process     | Function (Office)   | Rule                     | leads to                 |
| Office process     | Function (Office)   | CD-ROM                   | creates output to        |
| Office process     | Function (Office)   | Hard disk                | creates output to        |
| Office process     | Function (Office)   | Wastepaper basket        | creates output to        |
| Office process     | Function (Office)   | File cabinet             | creates output to        |
| Office process     | Function (Office)   | File                     | creates output to        |
| Office process     | Function (Office)   | Document                 | creates output to        |
| Office process     | Function (Office)   | Letter                   | creates output to        |
| Office process     | Function (Office)   | Diskette                 | creates output to        |
| Office process     | Function (Office)   | Book                     | creates output to        |
| Office process     | Function (Office)   | Time planner             | creates output to        |
| Office process     | Function (Office)   | Notepad                  | creates output to        |
| Office process     | Function (Office)   | Magnetic tape            | creates output to        |
| Office process     | Function (Office)   | Internet                 | creates output to        |
| Office process     | Function (Office)   | Fax                      | creates output to        |
| Office process     | Function (Office)   | Telephone                | creates output to        |
| Office process     | Function (Office)   | E-mail                   | creates output to        |
| Office process     | Function (Office)   | File bin                 | creates output to        |
| Office process     | Function (Office)   | Printer                  | creates output to        |
| Office process     | Function (Office)   | Location                 | is executed at           |



|                      |                     |                     |                             |
|----------------------|---------------------|---------------------|-----------------------------|
| Office process       | Rule                | Event               | leads to                    |
| Office process       | Rule                | Function (Office)   | activates                   |
| Office process       | Rule                | Rule                | links                       |
| Office process       | CD-ROM              | Function (Office)   | provides input for          |
| Office process       | Hard disk           | Function (Office)   | provides input for          |
| Office process       | Wastepaper basket   | Function (Office)   | provides input for          |
| Office process       | File cabinet        | Function (Office)   | provides input for          |
| Office process       | File                | Function (Office)   | provides input for          |
| Office process       | Document            | Function (Office)   | provides input for          |
| Office process       | Letter              | Function (Office)   | provides input for          |
| Office process       | Diskette            | Function (Office)   | provides input for          |
| Office process       | Book                | Function (Office)   | provides input for          |
| Office process       | Time planner        | Function (Office)   | provides input for          |
| Office process       | Notepad             | Function (Office)   | provides input for          |
| Office process       | Magnetic tape       | Function (Office)   | provides input for          |
| Office process       | Internet            | Function (Office)   | provides input for          |
| Office process       | Fax                 | Function (Office)   | provides input for          |
| Office process       | Telephone           | Function (Office)   | provides input for          |
| Office process       | E-mail              | Function (Office)   | provides input for          |
| Office process       | File bin            | Function (Office)   | provides input for          |
| Office process       | Printer             | Function (Office)   | provides input for          |
| Office process       | Computer            | Function (Office)   | can support                 |
| Office process       | Person (m)          | Function (Office)   | executes                    |
| Office process       | Person (m)          | Computer            | can be user                 |
| Office process       | Person (f)          | Function (Office)   | executes                    |
| Office process       | Person (f)          | Computer            | can be user                 |
| Office process       | Person type         | Function (Office)   | executes                    |
| Office process       | Person type         | Computer            | can be user                 |
| Office process       | Organizational unit | Function (Office)   | executes                    |
| Office process       | Organizational unit | Computer            | can be user                 |
| Office process       | Group               | Function (Office)   | executes                    |
| Office process       | Group               | Computer            | can be user                 |
| Office process       | Position            | Function (Office)   | executes                    |
| Office process       | Position            | Computer            | can be user                 |
| Organizational chart | Organizational unit | Organizational unit | is superior                 |
| Organizational chart | Organizational unit | Internal person     | belongs to                  |
| Organizational chart | Organizational unit | External person     | belongs to                  |
| Organizational chart | Organizational unit | Position            | is composed of              |
| Organizational chart | Organizational unit | Location            | is located at               |
| Organizational chart | Internal person     | Organizational unit | is Organization Manager for |
| Organizational chart | Internal person     | Internal person     | is Organization Manager for |
| Organizational chart | Internal person     | External person     | is Organization Manager for |
| Organizational chart | Internal person     | Position            | is Organization Manager for |



|                           |                   |                     |                             |
|---------------------------|-------------------|---------------------|-----------------------------|
| Organizational chart      | Internal person   | Location            | is located at               |
| Organizational chart      | Internal person   | Group               | is Organization Manager for |
| Organizational chart      | External person   | Organizational unit | belongs to                  |
| Organizational chart      | External person   | Internal person     | is Organization Manager for |
| Organizational chart      | External person   | External person     | is Organization Manager for |
| Organizational chart      | External person   | Position            | is Organization Manager for |
| Organizational chart      | External person   | Location            | is located at               |
| Organizational chart      | Position          | Organizational unit | is Organization Manager for |
| Organizational chart      | Position          | Internal person     | is Organization Manager for |
| Organizational chart      | Position          | External person     | is Organization Manager for |
| Organizational chart      | Position          | Position            | is Organization Manager for |
| Organizational chart      | Position          | Location            | is Organization Manager for |
| Organizational chart      | Position          | Group               | is Organization Manager for |
| Organizational chart      | Location          | Location            | subsumes                    |
| Organizational chart      | Group             | Organizational unit | is assigned to              |
| Organizational chart      | Group             | Internal person     | has member                  |
| Organizational chart      | Group             | External person     | has member                  |
| Organizational chart      | Group             | Position            | is composed of              |
| Organizational chart      | Group             | Location            | is located at               |
| Organizational chart      | Group             | Group               | cooperates with             |
| Organizational chart      | Person type       | Organizational unit | belongs to                  |
| Organizational chart      | Person type       | Position            | occupies                    |
| Technical terms model     | Technical term    | Technical term      | has relation with           |
| Technical terms model     | Technical term    | Cluster             | depicts                     |
| Value added chain diagram | Value-added chain | Value-added chain   | is predecessor of           |
| Value added chain diagram | Value-added chain | Value-added chain   | is predecessor of           |
| Value added chain diagram | Value-added chain | Technical term      | has output of               |
| Value added chain diagram | Value-added chain | Cluster             | has output of               |
| Value added chain diagram | Value-added chain | Value-added chain   | is predecessor of           |
| Value added chain diagram | Value-added chain | Value-added chain   | is predecessor of           |
| Value added chain diagram | Value-added chain | Technical term      | has output of               |

|                           |                     |                   |               |
|---------------------------|---------------------|-------------------|---------------|
| Value added chain diagram | Value-added chain   | Cluster           | has output of |
| Value added chain diagram | Technical term      | Value-added chain | is input for  |
| Value added chain diagram | Technical term      | Value-added chain | is input for  |
| Value added chain diagram | Organizational unit | Value-added chain | executes      |
| Value added chain diagram | Organizational unit | Value-added chain | executes      |
| Value added chain diagram | Cluster             | Value-added chain | is input for  |
| Value added chain diagram | Cluster             | Value-added chain | is input for  |

## ПРИЛОЖЕНИЕ 4

### Отчет о семантической проверке модели eEPC

#### Семантическая проверка ARIS

Сервер: LOCAL

БД: easydb

Пользователь: system

Правило: Каждый путь должен начинаться и заканчиваться событием.

Описание: Данное правило проверяет, все ли пути выбранной модели начинаются и заканчиваются объектом типа событие.

The following start or target objects are not events (Следующие начальные и конечные объекты – не события):

| <b>Checked model</b><br><b>Проверяемая модель</b> | <b>Object name</b><br><b>Имя объекта</b> | <b>Object type</b><br><b>Тип объекта</b> |
|---|--|--|
| 1. Первый тестовый пример                         | Функция 1                                | Function                                 |
|   | Функция 3                                | Function                                 |

Каждый путь должен начинаться и заканчиваться событием.

Правило: Количество входящих и исходящих связей оператора.

Описание: Данное правило проверяет, существует ли при каждом простом операторе выбранной модели только одна входящая и, как минимум, две исходящие связи, или, как минимум, две входящих и только одна исходящая связь.

The number of incoming and outgoing connections is not correct for the following operators (Число входящих и исходящих связей неправильно для следующих операторов):

| <b>Checked model</b><br><b>Проверяемая модель</b> | <b>Object name</b><br><b>Имя объекта</b> | <b>Object type</b><br><b>Тип объекта</b> |
|---|--|--|
| 1. Первый тестовый пример                         | Функция 2 (предшественник)               | Function                                 |
|   | OR operator                              | Rule                                     |
|   | Функция 3 (последователь)                | Function                                 |

Количество входящих и исходящих связей оператора.

Правило: Все функции и события имеют только одну входящую и исходящую связь.

Описание: Данное правило проверяет, имеют ли все функции и события выбранной модели не более одной входящей или исходящей связи.

The following functions and events have more than one incoming or outgoing connection (Следующие функции и события имеют больше чем одну входящую или исходящую связь):

| <b>Checked model</b><br><b>Проверяемая модель</b> | <b>Object name</b><br><b>Имя объекта</b> | <b>Object type</b><br><b>Тип объекта</b> |
|---|--|--|
| 1. Первый тестовый пример                         | Функция 4                                | Function                                 |

Все функции и события имеют только одну входящую и исходящую связь.

Правило: Невозможность XOR/OR после события.

Описание: Данное правило проверяет наличие оператора XOR или OR в процессе после события. Такие события при семантической проверке объявляются нарушением правила.

The following events have as successor an OR or an XOR operator (Следующие события имеют после себя оператор OR или XOR):

| <i>Checked model</i><br><i>Проверяемая модель</i>  | <i>Event</i><br><i>Событие</i> | <i>Following operator</i><br><i>Последующий оператор</i> |
|--|--------------------------------|--|
| Check produced no errors.<br>Ошибки не обнаружены. |                                |  |

Невозможность XOR/OR после события.

Правило: Должен сохраняться порядок при операторе.

Описание: Данное правило проверяет, не налагаются ли друг на друга набор типов предшествующих объектов и набор типов последующих объектов для объекта типа правило. Например, в тех случаях, когда все входящие в логический оператор связи исходят из событий, тогда все исходящие связи могут идти только к функциям и наоборот.

The following operators have the same object types both as predecessors and as succesors (Следующие операторы имеют тот же набор последующих типов объектов, что и набор предшествующих типов объектов):

| <i>Checked model</i><br><i>Проверяемая модель</i> | <i>Object name</i><br><i>Имя объекта</i> | <i>Object type</i><br><i>Тип объекта</i> |
|---|--|--|
| 1. Первый тестовый пример                         | Функция 1 (предшественник)               | Function                                 |
|   | XOR operator                             | Rule                                     |
|   | Событие 2 (последователь)                | Event                                    |
|   | Функция 2 (последователь)                | Function                                 |
|   | Функция 2 (предшественник)               | Function                                 |
|   | OR operator                              | Rule                                     |
|   | Функция 3 (последователь)                | Function                                 |

Должен сохраняться порядок при операторе.

| <i>Rule(правило)/Models № (модель №)</i>                              | <i>1</i> |
|---|----------|
| 1. Каждый путь должен начинаться и заканчиваться событием             | 2        |
| 2. Количество входящих и выходящих связей оператора                   | 3        |
| 3. Все функции и события имеют только одну входящую и выходящую связь | 1        |
| 4. Невозможность XOR/OR после события                                 | 0        |
| 5. Должен сохраняться порядок при операторе                           | 2        |

Статистика – правила.

## ПРИЛОЖЕНИЕ 5

### Отчет о семантической проверке организационной диаграммы

#### Семантическая проверка ARIS

Сервер: LOCAL

БД: easydb

Пользователь: system

Правило: Возможен только один корень.

Описание: Данное правило проверяет, содержат ли иерархические модели более одного объекта структурно-релевантного типа объекта, который не имеет входящей связи структурно-релевантного типа отношений.

The following models have more than one root (Следующие модели имеют больше чем один корень):

| <b>Checked model</b><br><b>Проверяемая модель</b>  | <b>Type</b><br><b>Тип</b> |
|--|---------------------------|
| Check produced no errors.<br>Ошибки не обнаружены. |                           |

Возможен только один корень.

Правило: Каждый объект может иметь только одного непосредственного предшественника.

Описание: Данное правило проверяет, содержит ли иерархическая модель объект, имеющий два или более главенствующих объекта, т.е. более одной входящей связи структурно-релевантного типа отношения.

The following objects have more than one direct predecessor (Следующие объекты имеют больше чем одного прямого предшественника):

| <b>Checked model</b><br><b>Проверяемая модель</b> | <b>Incorrect object</b><br><b>Неправильный объект</b> | <b>Object type</b><br><b>Тип объекта</b> |
|---|---|--|
| 1. Тестовый пример №2                             | Сотрудник отд.1 №2                                    | Position                                 |

Каждый объект может иметь только одного непосредственного предшественника.

Правило: Все соединения в модели должны быть одного типа.

Описание: Данное правило проверяет, содержит ли иерархическая модель объекты структурно-релевантного типа объекта, которые были связаны соединениями других структурно-релевантных типов отношений.

Connections of another relationship type were found between the following objects (Подключения другого типа отношений были найдены между следующими объектами):

| <b>Checked model</b><br><b>Проверяемая модель</b> | <b>Source object</b><br><b>Объект источник</b> | <b>Target object</b><br><b>Целевой объект</b> |
|---|--|---|
| 1. Тестовый пример №2                             | Отдел №2 (Organizational unit)                 | Сотрудник отд.2 №2 (Position)                 |
| 1. Тестовый пример №2                             |  | Сотрудник отд.1 №2 (Position)                 |
| 1. Тестовый пример №2                             |  | Сотрудник отд.2 №1 (Position)                 |
|   | Отдел №1 (Organizational unit)                 | Сотрудник отд.1 №1 (Position)                 |
|   |  | Сотрудник отд.1 №2 (Position)                 |

Все соединения в модели должны быть одного типа.

---

---

| <i>Rule(правило)/Models № (модель №)</i>                                     | <i>1</i> |
|--|----------|
| 1. Возможен только один корень   | 0        |
| 2. Каждый объект может иметь только одного непосредственного предшественника | 1        |
| 3. Все соединения в модели должны быть одного типа                           | 5        |

Статистика – правила.

## ПРИЛОЖЕНИЕ 6

### Отчет о семантической проверке функционального дерева

#### Семантическая проверка ARIS

Сервер: LOCAL

БД: easydb

Пользователь: system

Правило: Объекты без связей.

Описание: Данное правило производит проверку модели на предмет существования объектов без связи с другими объектами. Каждый объект должен иметь не менее одного предшественника и/или последователя.

The following objects have no connections to other objects (Следующие объекты не имеют связи с другими объектами):

| <i>Checked model</i><br><i>Проверяемая модель</i> | <i>Object name</i><br><i>Имя объекта</i> | <i>Object type</i><br><i>Тип объекта</i> |
|---|--|--|
| 1. Тестовый пример №3                             | Функция 8                                | Function                                 |

Объекты без связей.

Правило: Связи между двумя различными объектами.

Описание: Данное правило производит проверку модели на предмет соединения двух различных объектов в выбранной модели. Связь между двумя появлениями одного и того же объекта является ошибкой.

The following objects have a connection relationship to themselves (Следующие объекты имеют связи сами с собой):

| <i>Checked model</i><br><i>Проверяемая модель</i> | <i>Object name</i><br><i>Имя объекта</i> | <i>Object type</i><br><i>Тип объекта</i> |
|---|--|--|
| 1. Тестовый пример №3                             | Функция 2                                | Function                                 |

Связи между двумя различными объектами.

Правило: В модели не могут существовать циклы.

Описание: Данное правило проверяет все пути дерева на наличие циклов.

In the following models there are cycles in a path of the tree (В следующих моделях имеются циклы в пути дерева):

| <i>Checked model</i><br><i>Проверяемая модель</i> | <i>Type</i><br><i>Тип</i> |
|---|---------------------------|
| Тестовый пример №3                                | Function tree             |

В модели не могут существовать циклы.

Правило: Каждый объект может иметь только одного непосредственного предшественника.

Описание: Данное правило проверяет, содержит ли иерархия модели объект, имеющий два или более главенствующих объекта, т.е. более одной входящей связи структурно-релевантного типа отношения.

The following objects have more than one direct predecessor (Следующие объекты имеют больше чем одного прямого предшественника):

| <i>Checked model</i><br><i>Проверяемая модель</i> | <i>Object name</i><br><i>Имя объекта</i> | <i>Object type</i><br><i>Тип объекта</i> |
|---|--|--|
| 1. Тестовый пример №3                             | Функция 5                                | Function                                 |

Каждый объект может иметь только одного непосредственного предшественника.



| <i>Rule(правило)/Models № (модель №)</i>                                     | <i>1</i> |
|--|----------|
| 1. Объекты без связей  | 1        |
| 2. Связи между двумя различными объектами                                    | 1        |
| 3. В модели не могут существовать циклы                                      | 1        |
| 4. Каждый объект может иметь только одного непосредственного предшественника | 1        |

Статистика – правила.

## ПРИЛОЖЕНИЕ 7

### Отчет о семантической проверке модели технических терминов

#### Семантическая проверка ARIS

Сервер: LOCAL

БД: easydb

Пользователь: system

Правило: Объекты без связей.

Описание: Данное правило производит проверку модели на предмет существования объектов без связи с другими объектами. Каждый объект должен иметь не менее одного предшественника и/или последователя.

The following objects have no connections to other objects (Следующие объекты не имеют связи с другими объектами):

| <i>Checked model</i><br><i>Проверяемая модель</i> | <i>Object name</i><br><i>Имя объекта</i> | <i>Object type</i><br><i>Тип объекта</i> |
|---|--|--|
| 1. Тестовый пример №4                             | Кластер 3                                | Cluster/Data model                       |

Объекты без связей.

Правило: Связи между двумя различными объектами.

Описание: Данное правило производит проверку модели на предмет соединения двух различных объектов в выбранной модели. Связь между двумя случаями (появлениями) одного и того же объекта является ошибкой.

The following objects have a connection relationship to themselves (Следующие объекты имеют связи сами с собой):

| <i>Checked model</i><br><i>Проверяемая модель</i> | <i>Object name</i><br><i>Имя объекта</i> | <i>Object type</i><br><i>Тип объекта</i> |
|---|--|--|
| Ошибки не обнаружены.                             |  |  |


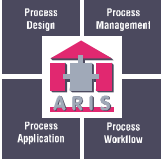
Связи между двумя различными объектами.

| <i>Rule/Models no.</i>                    | <i>1</i> |
|---|----------|
| 1. Объекты без связей                     | 1        |
| 2. Связи между двумя различными объектами | 0        |

Статистика – правила.

# ПРИЛОЖЕНИЕ 8

## Отчет №1 о функциях офисного процесса

|   |   |   |
|---|---|---|
|  | <p>Information about Model Functions<br/>Across Multiple Hierarchy Levels</p> |  |
|---|---|---|

### ARIS Report

Server: LOCAL

Database: AAA

User: system

### 1. Top-Level

#### 1.1 Прием КП eEPC

|               |                 |
|---------------|-----------------|
| <b>Group:</b> | <b>Процессы</b> |
| Attributes:   |                 |
| Name          | Прием КП        |

#### 1.1.1 Отправка КП в коммерческий отдел

|                       |                                  |
|-----------------------|----------------------------------|
| <b>Attributes:</b>    |                                  |
| Name                  | Отправка КП в коммерческий отдел |
| Avg. total costs      | 645,00 RUB                       |
| Avg. material costs   | 200,00 RUB                       |
| Avg. personnel costs  | 415,00 RUB                       |
| Avg. other costs      | 30,00 RUB                        |
| Avg. processing time  | 5,00 Minute(s)                   |
| Avg. orientation time | 5,00 Minute(s)                   |
| Avg. wait time        | 10,00 Minute(s)                  |
| Type                  | Function                         |
| Creator               | system                           |
| <b>Relationships:</b> |                                  |
| is activated by       | OR operator                      |
| Creates               | КП отправлено                    |
| is executed by        | Секретарь                        |

#### 1.1.2 Прием КП по факсу

|                       |  |
|-----------------------|--|
| <b>Attributes:</b>    |  |
| Name                  | Прием КП по факсу                          |
| Avg. total costs      | 1 014,50 RUB                               |
| Avg. material costs   | 400,00 RUB                                 |
| Avg. personnel costs  | 539,50 RUB                                 |
| Avg. other costs      | 75,00 RUB                                  |
| Avg. processing time  | 10,00 Minute(s)                            |
| Avg. orientation time | 10,00 Minute(s)                            |
| Avg. wait time        | 10,00 Minute(s)                            |
| Type                  | Function                                   |
| Creator               | system                                     |
| <b>Relationships:</b> |  |
| is executed by        | Секретарь                                  |
| Creates               | Факс принят                                |
| is activated by       | Решение контрагента о передаче КП по факсу |

## 1.1.3 Прием КП по электронной почте

|                       |   |
|-----------------------|---|
| <b>Attributes:</b>    |   |
| Name                  | Прием КП по эл. почте                         |
| Avg. total costs      | 1 595,50 RUB                                  |
| Avg. material costs   | 400,00 RUB                                    |
| Avg. personnel costs  | 1 120,50 RUB                                  |
| Avg. other costs      | 75,00 RUB                                     |
| Avg. processing time  | 15,00 Minute(s)                               |
| Avg. orientation time | 10,00 Minute(s)                               |
| Avg. wait time        | 5,00 Minute(s)                                |
| Type                  | Function                                      |
| Creator               | system  |
| <b>Relationships:</b> |   |
| is activated by       | Решение контрагента о передаче КП о эл. почте |
| is executed by        | Заместитель директора по ИТ                   |
| Creates               | Эл. КП отправлено                             |

## 1.1.4 Прием телефонного звонка от контрагента

|                       |   |
|-----------------------|---|
| <b>Attributes:</b>    |   |
| Name                  | Прием телефонного звонка от контрагента |
| Avg. total costs      | 645,00 RUB                              |
| Avg. material costs   | 200,00 RUB                              |
| Avg. personnel costs  | 415,00 RUB                              |
| Avg. other costs      | 30,00 RUB                               |
| Avg. processing time  | 10,00 Minute(s)                         |
| Avg. orientation time | 5,00 Minute(s)                          |
| Avg. wait time        | 10,00 Minute(s)                         |
| Type                  | Function                                |
| Creator               | system                                  |
| <b>Relationships:</b> |   |
| is executed by        | Секретарь                               |
| Creates               | Телефонный звонок принят                |
| is activated by       | Решение контрагента о связи потелефону  |

## 1.1.5 Регистрация КП


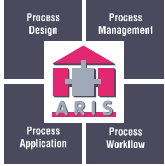
|                       |                          |
|-----------------------|--------------------------|
| <b>Attributes:</b>    |                          |
| Name                  | Регистрация КП           |
| Avg. total costs      | 1 697,50 RUB             |
| Avg. material costs   | 600,00 RUB               |
| Avg. personnel costs  | 1 037,50 RUB             |
| Avg. other costs      | 60,00 RUB                |
| Avg. processing time  | 20,00 Minute(s)          |
| Avg. orientation time | 15,00 Minute(s)          |
| Avg. wait time        | 10,00 Minute(s)          |
| Type                  | Function                 |
| Creator               | system                   |
| <b>Relationships:</b> |                          |
| Creates               | КП зарегистрировано      |
| is executed by        | Секретарь                |
| is activated by       | Эл. вариант КП составлен |

## 1.1.6 Составление эл. варианта КП

|                       |                             |
|-----------------------|-----------------------------|
| <b>Attributes:</b>    |                             |
| Name                  | Составление эл. варианта КП |
| Avg. total costs      | 852,50 RUB                  |
| Avg. material costs   | 200,00 RUB                  |
| Avg. personnel costs  | 622,50 RUB                  |
| Avg. other costs      | 30,00 RUB                   |
| Avg. processing time  | 20,00 Minute(s)             |
| Avg. orientation time | 10,00 Minute(s)             |
| Avg. wait time        | 10,00 Minute(s)             |
| Type                  | Function                    |
| Creator               | system                      |
| <b>Relationships:</b> |                             |
| is executed by        | Секретарь                   |
| is activated by       | OR operator                 |
| Creates               | Эл. вариант КП составлен    |

# ПРИЛОЖЕНИЕ 9

## Отчет №2 о функциях офисного процесса

|   |   |   |
|---|---|---|
|  | Information about Model Functions Across Multiple Hierarchy Levels which is Relevant for Certification. |  |
|---|---|---|

### ARIS Report

Server: LOCAL

Database: AAA

User: system

### 1. Top-Level

#### 1.1 Прием КП еЕPC

|               |          |
|---------------|----------|
| <b>Group:</b> | Процессы |
| Attributes:   |          |
| Name          | Прием КП |

#### 1.1.1 Отправка КП в коммерческий отдел

|                                       |                                  |
|---------------------------------------|----------------------------------|
| <b>Attributes:</b>                    |                                  |
| Name                                  | Отправка КП в коммерческий отдел |
| Avg. total costs                      | 645,00 RUB                       |
| Avg. material costs                   | 200,00 RUB                       |
| Avg. personnel costs                  | 415,00 RUB                       |
| Avg. other costs                      | 30,00 RUB                        |
| Avg. processing time                  | 5,00 Minute(s)                   |
| Avg. orientation time                 | 5,00 Minute(s)                   |
| Avg. wait time                        | 10,00 Minute(s)                  |
| Type                                  | Function                         |
| Creator                               | system                           |
| <b>Organizational responsibility:</b> |                                  |
| is executed by                        | Секретарь                        |

#### 1.1.2 Прием КП по факсу

|                                       |                   |
|---------------------------------------|-------------------|
| <b>Attributes:</b>                    |                   |
| Name                                  | Прием КП по факсу |
| Avg. total costs                      | 1 014,50 RUB      |
| Avg. material costs                   | 400,00 RUB        |
| Avg. personnel costs                  | 539,50 RUB        |
| Avg. other costs                      | 75,00 RUB         |
| Avg. processing time                  | 10,00 Minute(s)   |
| Avg. orientation time                 | 10,00 Minute(s)   |
| Avg. wait time                        | 10,00 Minute(s)   |
| Type                                  | Function          |
| Creator                               | system            |
| <b>Organizational responsibility:</b> |                   |
| is executed by                        | Секретарь         |

## 1.1.3 Прием КП по электронной почте

|                                       |                             |
|---------------------------------------|-----------------------------|
| <b>Attributes:</b>                    |                             |
| Name                                  | Прием КП по эл. почте       |
| Avg. total costs                      | 1 595,50 RUB                |
| Avg. material costs                   | 400,00 RUB                  |
| Avg. personnel costs                  | 1 120,50 RUB                |
| Avg. other costs                      | 75,00 RUB                   |
| Avg. processing time                  | 15,00 Minute(s)             |
| Avg. orientation time                 | 10,00 Minute(s)             |
| Avg. wait time                        | 5,00 Minute(s)              |
| Type                                  | Function                    |
| Creator                               | system                      |
| <b>Organizational responsibility:</b> |                             |
| Is executed by                        | Заместитель директора по ИТ |

## 1.1.4 Прием телефонного звонка от контрагента

|                                       |   |
|---------------------------------------|---|
| <b>Attributes:</b>                    |   |
| Name                                  | Прием телефонного звонка от контрагента |
| Avg. total costs                      | 645,00 RUB                              |
| Avg. material costs                   | 200,00 RUB                              |
| Avg. personnel costs                  | 415,00 RUB                              |
| Avg. other costs                      | 30,00 RUB                               |
| Avg. processing time                  | 10,00 Minute(s)                         |
| Avg. orientation time                 | 5,00 Minute(s)                          |
| Avg. wait time                        | 10,00 Minute(s)                         |
| Type                                  | Function                                |
| Creator                               | system                                  |
| <b>Organizational responsibility:</b> |   |
| Is executed by                        | Секретарь                               |

## 1.1.5 Регистрация КП

|                                       |                 |
|---------------------------------------|-----------------|
| <b>Attributes:</b>                    |                 |
| Name                                  | Регистрация КП  |
| Avg. total costs                      | 1 697,50 RUB    |
| Avg. material costs                   | 600,00 RUB      |
| Avg. personnel costs                  | 1 037,50 RUB    |
| Avg. other costs                      | 60,00 RUB       |
| Avg. processing time                  | 20,00 Minute(s) |
| Avg. orientation time                 | 15,00 Minute(s) |
| Avg. wait time                        | 10,00 Minute(s) |
| Type                                  | Function        |
| Creator                               | system          |
| <b>Organizational responsibility:</b> |                 |
| Is executed by                        | Секретарь       |



## 1.1.6 Составление эл. варианта КП

|                                       |                             |
|---------------------------------------|-----------------------------|
| <b>Attributes:</b>                    |                             |
| Name                                  | Составление эл. варианта КП |
| Avg. total costs                      | 852,50 RUB                  |
| Avg. material costs                   | 200,00 RUB                  |
| Avg. personnel costs                  | 622,50 RUB                  |
| Avg. other costs                      | 30,00 RUB                   |
| Avg. processing time                  | 20,00 Minute(s)             |
| Avg. orientation time                 | 10,00 Minute(s)             |
| Avg. wait time                        | 10,00 Minute(s)             |
| Type                                  | Function                    |
| Creator                               | system                      |
| <b>Organizational responsibility:</b> |                             |
| Is executed by                        | Секретарь                   |

**Organizational elements:**

Секретарь

|                    |             |
|--------------------|-------------|
| <b>Attributes:</b> |             |
| Name               | Секретарь   |
| Type               | Person type |
| Creator            | system      |

Заместитель директора по ИТ

|                    |                             |
|--------------------|-----------------------------|
| <b>Attributes:</b> |                             |
| Name               | Заместитель директора по ИТ |
| Type               | Person type                 |
| Creator            | system                      |

# ***СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ***

1. Август-Вильгельм Шеер.

**Бизнес-процессы. Основные понятия. Теория. Методы.**

М.: ООО "Издательство "Серебряные нити", 1999. – 152 с.

2. Август-Вильгельм Шеер.

**Моделирование бизнес-процессов.**

М.: Изд. АОЗТ "Просветитель", 2000. – 205 с.

3. Август-Вильгельм Шеер.

**"Основать-то компанию просто ...".** Пер. с нем.

М.: "Весть-Метатехнология", 2001. – 282 с.

4. Агафонов В.Н.

**Спецификация программ: понятийные средства и их организация.**

Новосибирск: Наука, 1987. – 234 с.

5. Буч Г.

**Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд.,**  
Пер. с англ.

М.: "Издательство Бином", СПб.: "Невский диалект", 1998. – 560 с.

6. Буч Г., Рамбо Д., Джекобсон А.

**Язык UML. Руководство пользователя.,** Пер. с англ.

М.: ДМК Пресс, 2001. – 432 с.

7. Вендов А.М.

**CASE-технологии. Современные методы и средства проектирования информационных систем.**

М.: Финансы статистика, 1998. – 176 с.

8. Вендов А.М.

**Проектирование программного обеспечения экономических информационных систем. Учебник.**

М.: Финансы и статистика, 2000. – 352 с.

9. Весин В.Р.

**Менеджмент. Учебное пособие в схемах.**

М.: Белые альвы, 1999. – 154 с.

10. Войнов И.В., Телегин А.И.

**Основы экономики и интегрированных информационных систем. Учебное пособие.**

Челябинск: Изд-во ЮУрГУ, 2000. – 114 с.

11. Войнов И.В., Телегин А.И.

**Математические модели в экономике. Учебное пособие.**

Челябинск: Изд-во ЮУрГУ, 2000. – 91 с.

12. Войнов И.В., Телегин А.И.

**Оптимальное управление экономическими системами. Учебное пособие.**

Челябинск: Изд-во ЮУрГУ, 1999. – 65 с.

13. Дорот В.А., Новиков Ф.А.

**Толковый словарь современной компьютерной лексики.**

СПб.: БХВ – Санкт-Петербург, 1999. – 384 с.

14. Ивлев В.А., Попова Т.В.

**Реорганизация деятельности предприятий: от структурной к процессной организации.**

М.: ООО Изд-во “Научтехлитиздат”, 2000. – 263 с.

15. Каменнова М., Громов А., Ферапонтов М., Шматалюк А.

**Моделирование бизнеса. Методология ARIS.**

М.: ООО “Издательство “Серебряные нити”, 2001. – 327 с.

16. Каморджанова Н.А., Карташова И.В.

**Бухгалтерский учет в схемах и рисунках.** Учеб. пособие.

М.: ИНФРА-М, 2000. – 429 с.

17. Калянов Г.Н.

**CASE структурный системный анализ (автоматизация и применение).**

М.: Изд. “ЛОРИ”, 1996. – 242 с.

18. Кравченко В.Ф., Кравченко Е.Ф., Забелин П.В.

**Организационный инжиниринг.** Учебное пособие.

М.: “Издательство ПРИОР”, 1999. – 25с.

19. Леоненков А.В.

**Самоучитель UML.**

СПб.: БХВ-Петербург, 2001. – 304 с.

20. Райзенберг Б.А., Лозовский Л.Ш., Стародубцев Е.Б.

**Современный экономический словарь.** 2-е изд., исправ.

М.: ИНФРА-М, 1999. – 479 с.

21. Скрипкин К.Г.

**Финансовая информатика.** Учебное пособие.

М.: ТЕИС, 1997. – 160 с.

22. **Справочник моделей ARIS с примерами. Часть 1.**

М.: VIP Anatech, 2000. – 122 с.

23. **Справочник моделей ARIS с примерами. Часть 2.**

М.: VIP Anatech, 2000. – 118 с.

24. Фаулер М., Скотт К.

**UML в кратком изложении. Применение стандартного языка объектного моделирования.**

М.: Мир, 1999. – 191 с.

25. Фишер Лэйна.

**Совершенство на практике. Лучшие проекты в области управления бизнес-процессами и workflow.**

Пер. с нем.

М.: “Весть-Метатехнология”, 2000. – 432 с.

26. Хаммер М., Чампи Дж.

**Реинжиниринг корпорации: Манифест революции в бизнесе.** Пер. с англ.

СПб.: Издательство С.-Петербургского университета, 1997. – 332 с.

27. Четыркин Е.М.

**Финансовый анализ производственных инвестиций.**

М.: Дело, 1998. – 256 с.

28. Шлеер С., Меллор С.

**Объектно-ориентированный анализ: Моделирование мира в состояниях.**

Киев: Диалектика, 1993. – 240 с.

29. **Экономикс.** Англо-русский словарь-справочник.

Э.Дж. Долан, Б.И. Домненко.

М.: Лазурь, 1994. – 544 с.

# ОГЛАВЛЕНИЕ

|  |           |
|--|-----------|
| <b>ВВЕДЕНИЕ .....</b>  | <b>3</b>  |
| Зачем нужны модели и моделирование? .....  | 3         |
| Почему модели имеют графический (диаграммный) вид? .....                             | 3         |
| Где применяются CASE-средства? .....   | 4         |
| Для чего используются модели? .....  | 4         |
| Какие особенности у функционального и объектно-ориентированного моделирования? ...   | 5         |
| Почему выбирают АРИС-моделирование? .....  | 6         |
| Чем хороша программная среда ARIS 5.0? .....   | 7         |
| Что изложено в этой книге? .....   | 8         |
| Кому рекомендуется эта книга? .....  | 9         |
| Что мы использовали и кто нам помог? .....   | 10        |
| <br><b>1. МОДЕЛИРОВАНИЕ ОСНОВ ЭКОНОМИЧЕСКИХ СИСТЕМ И ПРОЦЕССОВ .....</b>             | <b>12</b> |
| 1.1 Системы и их классификация .....   | 12        |
| 1.2 Экономические системы .....  | 14        |
| 1.3 Производственные ресурсы (средства производства) .....                           | 15        |
| 1.4 Производственные процессы .....  | 18        |
| 1.5 Основные понятия финансовых подсистем .....                                      | 24        |
| <br><b>2. ИНФОРМАТИКА В ЭКОНОМИКЕ .....</b>  | <b>33</b> |
| 2.1 Некоторые понятия информатики .....  | 33        |
| 2.2 Экономическая информация в управлении .....                                      | 35        |
| 2.3 Прикладная информатика в экономике .....   | 39        |
| 2.4 Конфиденциальная информация и ее защита .....                                    | 43        |
| 2.5 Офис как информационное предприятие .....  | 47        |
| 2.6 Распределенные информационные системы .....                                      | 51        |
| <br><b>3. ПРОЦЕССНО-ОРИЕНТИРОВАННОЕ УПРАВЛЕНИЕ .....</b>                             | <b>56</b> |
| 3.1 Функционально-ориентированное управление .....                                   | 56        |
| 3.2 Бизнес-процессы .....  | 57        |
| 3.3 Управление бизнес-процессом .....  | 59        |
| 3.4 Управление потоками работ .....  | 61        |
| 3.5 Этапы перехода к процессному управлению .....                                    | 62        |
| 3.6 Инжиниринг и оптимизация бизнес-процессов .....                                  | 65        |
| 3.7 Основные направления реинжиниринга .....   | 66        |
| 3.8 Характеристики перепроектированных бизнес-процессов .....                        | 68        |
| <br><b>4. ОБСЛЕДОВАНИЕ И ВЕРБАЛЬНОЕ ОПИСАНИЕ СИСТЕМ .....</b>                        | <b>70</b> |
| 4.1 Обследование предприятия .....   | 70        |
| 4.2 Сбор информации о предприятии .....  | 71        |
| 4.3 Вербальное описание систем .....   | 72        |
| 4.4 Этапы вербального описания предприятий .....                                     | 73        |
| <br><b>5. АРХИТЕКТУРА ИНТЕГРИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ<br/>    (АРИС) .....</b> | <b>76</b> |
| 5.1 АРИС-методология моделирования систем .....                                      | 76        |
| 5.2 Некоторые особенности АРИС-моделирования .....                                   | 77        |
| 5.3 Этапы описания системы .....   | 79        |
| 5.4 Виды и типы моделей .....  | 82        |
| 5.5 Уровни моделирования .....   | 86        |

|  |            |
|--|------------|
| 5.6 Детализация и структуризация моделей .....                       | 100        |
| 5.7 Управление знаниями .....  | 103        |
| <b>6. ОПИСАНИЕ ПРОГРАММНОЙ СРЕДЫ ARIS .....</b>                      | <b>106</b> |
| 6.1 Структура и возможности .....                                    | 106        |
| 6.2 Основные свойства моделей и объектов .....                       | 112        |
| <i>Model Appearance</i> .....  | 114        |
| <i>Attributes</i> .....  | 115        |
| <i>Maintained Attributes</i> .....                                   | 116        |
| <i>Relationships</i> .....   | 118        |
| <i>Attributes Placements</i> .....                                   | 118        |
| <i>Assignments</i> .....   | 118        |
| <i>Object Appearance</i> .....                                       | 118        |
| 6.3 Организационное моделирование .....                              | 122        |
| 6.4 Моделирование данных .....                                       | 134        |
| 6.5 Функциональное моделирование .....                               | 144        |
| 6.6 Процессный вид моделирования .....                               | 154        |
| <i>Модель типа Communications diagram</i> .....                      | 154        |
| <i>Модель типа Value-added chain diagram</i> .....                   | 155        |
| <i>Модель типа EPCs</i> .....  | 160        |
| Другие типы моделей .....  | 165        |
| Этапы спецификации и реализации системы и ее БП .....                | 173        |
| Поддерживаемые методы моделирования и ARIS-интеграция .....          | 176        |
| 6.7 Моделирование знаний .....                                       | 179        |
| 6.8 Использование моделей .....                                      | 181        |
| Скрипты отчетов по объектам .....                                    | 190        |
| Скрипты отчетов по БД .....  | 190        |
| Скрипты отчетов по папкам БД .....                                   | 190        |
| Скрипты отчетов по таблицам функционально-стоимостного анализа ..... | 191        |
| Скрипт отчета по фильтрам .....                                      | 191        |
| 6.9 Семантический контроль .....                                     | 191        |
| <b>7. ARIS-МОДЕЛИРОВАНИЕ С ЛЕГКИМ ФИЛЬТРОМ .....</b>                 | <b>202</b> |
| 7.1 Описание атрибутов моделей, объектов и связей .....              | 202        |
| 7.2 Типы моделей и примеры их построения .....                       | 202        |
| 7.3 Формирование ARIS-отчетов по моделям .....                       | 221        |
| 7.4 Скрипты анализа моделей и создание новых скриптов .....          | 231        |
| 7.5 Программы обработки моделей технических терминов .....           | 237        |
| 7.6 Программа анализа функционального дерева .....                   | 242        |
| 7.7 Программа обработки организационной диаграммы .....              | 242        |
| 7.8 Программа “прокрутки” eEPC-модели .....                          | 248        |
| <b>8. ПРИМЕРЫ МОДЕЛИРОВАНИЯ СИСТЕМ И ПРОЦЕССОВ .....</b>             | <b>255</b> |
| 8.1 Обслуживание клиентов гостиницы .....                            | 255        |
| 8.2 Услуги интернет-салона .....                                     | 255        |
| 8.3 Телевизионные игры .....   | 260        |
| 8.4 Разработка технических документов .....                          | 269        |
| 8.5 Конкурсный отбор и виртуальная кафедра .....                     | 276        |
| 8.6 Офисная деятельность фирмы .....                                 | 293        |

---

|   |            |
|---|------------|
| <b>9. МОДЕЛИРОВАНИЕ ПРОИЗВОДСТВЕННО-СБЫТОВОЙ ФИРМЫ .....</b>                          | <b>317</b> |
| 9.1 Основное балансовое уравнение .....   | 317        |
| 9.2 Вычисление выручки и каноническая форма балансового уравнения .....               | 318        |
| 9.3 Вычисление объема производства и цены продукции .....                             | 319        |
| 9.4 Моделирование объемов выпуска продукции .....                                     | 321        |
| 9.5 Вычисление затрат .....   | 323        |
| 9.6 Сценарий моделирования финансово-обеспеченного производства .....                 | 326        |
| 9.7 Моделирование фирмы по производству и реализации мужских сорочек .....            | 330        |
| 9.8 Особенности дискретного динамического моделирования .....                         | 331        |
| <b>ЗАКЛЮЧЕНИЕ .....</b>   | <b>339</b> |
| <b>ПРИЛОЖЕНИЕ 1. Таблица типов моделей ARIS 5.0 с полным фильтром .....</b>           | <b>340</b> |
| <b>ПРИЛОЖЕНИЕ 2. Краткое описание основных ARIS-объектов .....</b>                    | <b>348</b> |
| <b>ПРИЛОЖЕНИЕ 3. Таблица типов отношений .....</b>                                    | <b>360</b> |
| <b>ПРИЛОЖЕНИЕ 4. Отчет о семантической проверке модели eEPC .....</b>                 | <b>371</b> |
| <b>ПРИЛОЖЕНИЕ 5. Отчет о семантической проверке организационной диаграммы .....</b>   | <b>373</b> |
| <b>ПРИЛОЖЕНИЕ 6. Отчет о семантической проверке функционального дерева .....</b>      | <b>375</b> |
| <b>ПРИЛОЖЕНИЕ 7. Отчет о семантической проверке модели технических терминов .....</b> | <b>377</b> |
| <b>ПРИЛОЖЕНИЕ 8. Отчет №1 о функциях офисного процесса .....</b>                      | <b>378</b> |
| <b>ПРИЛОЖЕНИЕ 9. Отчет №2 о функциях офисного процесса .....</b>                      | <b>381</b> |
| <b>СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....</b>   | <b>384</b> |



## ОБ АВТОРАХ ЭТОЙ КНИГИ



**Войнов Игорь Вячеславович** – д.т.н., профессор, директор филиала Южно-Уральского государственного университета (ЮУрГУ) в г. Миассе, академик. Большой практический опыт приобрел, работая на инженерных, научных и руководящих должностях в НПО “Электромеханика”, ОАО “Уральский автомобильный завод”, ЮУрГУ и администрации города Миасса. В 2000 году избран депутатом законодательного собрания Челябинской области. Автор более пятидесяти научных публикаций, в том числе двадцати пяти патентов и авторских свидетельств на изобретения в области систем управления, робототехники и экономики.

**Пудовкина Светлана Геннадьевна** – преподаватель кафедры “Системы управления и математическое моделирование” (СУиММ) ЮУрГУ. Окончила с отличием Всероссийский заочный финансово-экономический институт по специальности “Бухгалтерский учет, аудит и анализ хозяйственной деятельности”. Работала главным бухгалтером в Миасском педагогическом колледже, а также в торговой и строительной фирмах. Специализируется в области математического моделирования и информационного обеспечения экономических систем и управления экономическими процессами.



**Телегин Александр Иванович** – д.ф.-м.н, профессор, декан электротехнического факультета ЮУрГУ, заведующий кафедрой СУиММ. Защитил кандидатскую и докторскую диссертации в Московском государственном университете. Специалист прикладной математики и информатики в области сложных систем и процессов. Автор десяти книг (пяти монографий и пяти учебных пособий) и более 40 научных статей. Имеет авторские свидетельства и патенты на изобретения в области робототехники. С 90-х годов прошлого столетия осваивает новые информационные технологии в экономике, в том числе ARIS-методологию и соответствующие программные продукты для моделирования, анализа, реинжиниринга и оптимизации экономических систем, а также оптимизации управления экономическими процессами.











Игорь Вячеславович Войнов,  
Светлана Геннадьевна Пудовкина,  
Александр Иванович Телегин

Моделирование экономических систем и процессов.  
Опыт построения ARIS-моделей.

Монография

Технический редактор А. В. Миних

Издательство Южно-Уральского государственного университета

ИД №00200 от 28.09.99г.

---

Подписано в печать 20.02.2002г.

Формат 60х84 1/8. Печать лазерная. Усл. печ. л. 45,57. Уч.\_изд. л. 49,12.

Тираж 300 экз. Заказ 293. Цена договорная.

---