

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Ю.Б. Гриценко

АРХИТЕКТУРА ПРЕДПРИЯТИЯ

Учебное пособие

*Рекомендовано
учебно-методическим объединением вузов РФ
в области экономики, менеджмента, логистики
и бизнес-информатики в качестве учебного пособия
для студентов высших учебных заведений, обучающихся
по направлению подготовки 080500 «Бизнес-информатика»*

Томск
Издательство ТУСУРа
2014

УДК 004.9:658
ББК 32.973.2-018
Г 85

Рецензенты:

Мельник А.Н., д-р экон. наук, профессор,
зав. кафедрой инноваций и инвестиций ФГАОУ ВПО
«Казанский (Приволжский) федеральный университет»

Сущенко С.П., д-р техн. наук, профессор,
зав. кафедрой прикладной информатики
Томского государственного университета

Гриценко, Юрий Борисович.

Г 85 Архитектура предприятия: учеб. пособие / Ю.Б. Гриценко. — Томск :
Изд-во Томск. гос. ун-та систем управления и радиоэлектроники, 2014. —
260 с.

ISBN 978-5-86889-512-8

Рассмотрено понятие предприятия, сформулировано определение архитектуры предприятия, представлены методологии построения архитектуры предприятия, а также модели и методики разработки архитектур предприятия, используемые в современном мире.

Предназначено для использования в качестве учебного пособия по дисциплине «Архитектура предприятия», изучаемой студентами направления 080500. 62 «Бизнес-информатика».

УДК 004.9:658
ББК 32.973.2-018

ISBN 978-5-86889-512-8

© Гриценко Ю.Б., 2014
© Томск. гос. ун-т систем управления
и радиоэлектроники, 2014

Предисловие

Необходимость в подготовке высококвалифицированных специалистов, способных эффективно заниматься организацией сложных информационных систем в бизнесе и государственном управлении, обуславливает дальнейшее развитие и совершенствование такого направления подготовки, как «Бизнес-информатика». В образовательном стандарте в качестве основных объектов профессиональной деятельности магистра бизнес-информатики отмечена архитектура информационных систем, которая является составной частью дисциплины «Архитектура предприятия». В рамках магистерской программы «Моделирование и оптимизация бизнес-процессов» по данному направлению предлагается к изучению дисциплина «Архитектура предприятия», не включенная в обязательный минимум основной образовательной программы, но освоение которой существенно дополнит знания выпускников в области ИТ-консалтинга, проектирования, внедрения и управления сложными информационными системами, менеджмента проектов.

Автором обобщен и систематизирован представленный в работах российских и зарубежных специалистов обширный материал по построению архитектуры предприятий, сформулировано понятие архитектуры предприятия с комплексным раскрытием составляющих его элементов, а также приведены классические и современные модели, методы и методики построения архитектур предприятия. Особо хотелось бы подчеркнуть присутствие в учебном пособии индивидуальных заданий, позволяющих по мере их выполнения построить полнофункциональную архитектуру предприятия из какой-либо выбранной области.

В условиях острого дефицита комплексных качественных обучающих материалов по данной тематике предлагаемое учебное пособие может значительно облегчить изучение дисциплины «Архитектура предприятия».

*Профессор, д-р техн. наук
В.Ф. Тарасенко*

Введение

При анализе аспектов, связанных с изменениями в бизнесе организации, традиционным является подход «сверху вниз» в терминах людей и процессов. Однако в сложившейся практике при разработке приложений и даже комплексных информационных систем (ИС) основные усилия направлены на идентификацию и реализацию специфических, функциональных свойств, которые требуются для автоматизации отдельных задач, т. е. движение в обратном направлении. Существенно меньше внимания уделяется взаимодействию системы в целом с другими системами в целях достижения преимуществ для бизнеса организации. В результате возникает несоответствие между высокоуровневым бизнес-видением, существующей организационной структурой и информационными системами, действующими на предприятии [1]. Для ликвидации этой ситуации многие организации начинают рассматривать **архитектуру предприятия** в целом. Именно такой подход и позволяет обеспечить единое видение бизнес-процессов организации. Эффективная архитектура предприятия характеризуется целостным и всеобъемлющим представлением как статического, так и динамического аспектов архитектуры, которые содержат следующие элементы:

- миссию предприятия и стратегическое планирование его деятельности;
- организационные структуры и сервисы, необходимые для реализации бизнес-архитектуры, позволяющей в полной мере удовлетворить требования миссии и стратегического планирования;
- системную архитектуру, которая требуется для эффективной реализации сервисов уровня бизнес-архитектуры.

Материал учебного пособия построен на базе работ в области ИТ-бизнеса таких авторов, как Г.Г. Верников, В.И. Галактионов, А.В. Данилин, Г.Н. Калянов, А.И. Слюсаренко и др., а также анализа описаний продуктов по построению архитектуры предприятия и руководств по их использованию различных фирм-производителей.

В первой главе учебного пособия раскрывается смысл таких терминов, как «предприятие», «архитектура предприятия»; подробно описываются статический и динамический аспекты построения архитектуры предприятия; рассматривается архитектура предприятия в разрезе различных слоев и показывается ее значение в различных условиях.

При рассмотрении термина «архитектура предприятия» раскрываются миссия и стратегическое планирование деятельности предприятия. Отмечается наличие на предприятии тактического и оперативного планирования, относящихся к среднесрочному и краткосрочному планированию.

Для построения архитектур предприятия специалисты могут использовать различные системные подходы. К ним относятся классические методологии построения сложных систем¹ на основе структурного и объектно-ориентированного анализа.

Вторая глава пособия содержит подробное описание методологий структурного и объектно-ориентированного анализа сложных систем. Наиболее популярными методологиями структурного анализа являются:

- методология на основе диаграмм потоков данных DFD;
- методология структурного анализа и проектирования IDEF0, входящая в группу методологий SADT;
- методология моделирования и стандарт документирования процессов IDEF3;
- методология моделирования отношений между данными IDEF1X.

¹ Процесс построения архитектуры предприятия является, бесспорно, процессом построения архитектуры сложной системы.

В качестве объектно-ориентированных методологий анализа и проектирования сложных систем предлагается унифицированный язык моделирования UML, возникший как результат обобщения работ в области объектно-ориентированного моделирования таких авторов, как Г. Буч, Дж. Рамбо и А. Якобсон. UML представляет собой нотацию для визуального моделирования программных систем и бизнес-процессов. В то же время описание языка UML не содержит сведений относительно того, каким образом и в какой последовательности следует разрабатывать канонические диаграммы при выполнении конкретных проектов.

Целью второй главы является не изучение методологий, а систематизация знаний студентов в области структурного и объектно-ориентированного анализа, полученных в предшествующих курсах, таких как «Объектно-ориентированный анализ и программирование», «Моделирование и анализ бизнес-процессов», «Проектирование информационных систем».

Моделирование реальных ситуаций в работе предприятий и отработка комплексных бизнес-процессов с использованием различных методов моделирования сопровождается значительными сложностями при практическом применении. Вследствие этого разработчики архитектур предприятия пытаются стандартизировать концепции построения архитектуры информационных систем и методов моделирования.

На современном рынке ведется агрессивная маркетинговая политика различных фирм, предлагающих свои концепции для построения архитектуры предприятия. Одной из таких концепций является «Архитектура интегрированных информационных систем» (ARIS), в которой предлагается ряд преимуществ: возможность выбора методов моделирования и их интеграции с учетом особенностей моделируемого объекта; построение базы для управления сложными проектами, обеспечиваемое наличием структурных элементов, содержащих встроенные модели процедур для разработки интегрированных информационных систем. Организация архитектуры ARIS приведена *в третьей главе* пособия. Данная архитектура рассматривается в разрезе нескольких крупных моделей:

- организационной;
- функциональной;
- информационной;
- управляющей;
- модели ресурсов.

Первые системные разработки архитектуры предприятия появились в 80-х гг. XX в. Одной из наиболее известных методологий построения именно архитектуры предприятия, а не просто сложной системы является модель Захмана. В 1987 г. Джон Захман опубликовал полезную схему развития архитектуры информационной системы. «Захмановская» схема создает контекст для описания различных представлений архитектуры разрабатываемой системы. Эти представления соответствуют видению системы заказчиком, проектировщиком и разработчиком, причем в разрезе трех выбранных аспектов, отражающих точку зрения каждого из участников процесса разработки ИС. Данная методология постоянно развивалась и составила впоследствии основу других методологий, таких как TOGAF, NASCIO Architecture Toolkit, SAM, EAP и других, некоторые из которых рассмотрены в *четвертой главе* пособия. Здесь же приводится краткое сравнение различных методологий по построению архитектуры предприятия.

При изучении дисциплины «Архитектура предприятия» в соответствии с разработанным учебным планом предусмотрены практические занятия по построению архитектуры какого-либо конкретного предприятия. Практические занятия содержат девять заданий (*приложение 1*), в результате выполнения которых студент должен представить полное описание архитектуры выбранного объекта с использованием уже известных моделей и методик. Допускается использование собственной методики, но с обязательным обоснованием внесенных усовершенствований.

Глава 1

АРХИТЕКТУРА ПРЕДПРИЯТИЯ В РАЗЛИЧНЫХ АСПЕКТАХ

1.1. Сущность и базовые понятия дисциплины

1.1.1. Предприятие как объект изучения

Предприятие в соответствии со ст. 132 Гражданского кодекса Российской Федерации — это имущественный комплекс, используемый для осуществления предпринимательской деятельности. Предприятие как имущественный комплекс признается недвижимостью. Предприятие в целом или его часть могут быть объектом купли-продажи, залога, аренды и других сделок, связанных с установлением, изменением и прекращением вещных прав. В состав предприятия как имущественного комплекса входят все виды предназначенного для деятельности предприятия имущества (земельные участки, здания, сооружения, оборудование, инвентарь, сырье, продукция), права требования и долги, а также права на обозначения, индивидуализирующие предприятие и его продукцию, работы и услуги (коммерческое обозначение, товарные знаки, знаки обслуживания), и другие исключительные права, если иное не предусмотрено законом или договором.

В современном экономическом словаре **предприятие** трактуется как обособленная специализированная единица, основанием которой является профессионально организованный трудовой коллектив, способный с помощью имеющихся в его распоряже-

нии средств производства изготовить нужную потребителю продукцию (выполнить работы, оказать услуги) соответствующего значения, профиля и ассортимента [2].

Предприятия классифицируют по различным признакам:

- **по форме собственности:**
 - государственной;
 - муниципальной;
 - частной;
 - собственности общественных организаций;
 - иных форм собственности (смешанная собственность, собственность иностранных лиц, лиц без гражданства);
- **по масштабу:**
 - малые;
 - средние;
 - крупные;
- **по организационно-экономической форме:**
 - индивидуальные;
 - партнёрства (хозяйственные товарищества и общества);
 - корпорации (акционерные общества, госкорпорации);
- **по цели деятельности:**
 - коммерческие;
 - некоммерческие;
- **по отраслевому и функциональному виду деятельности:**
 - промышленные;
 - сельскохозяйственные;
 - строительные;
 - транспортные;
 - торговые;
 - производственно-торговые;
 - торгово-посреднические;
 - инновационно-внедренческие;
 - лизинговые;
 - банковские;
 - страховые;
 - туристические и др.;

• *по технологической целостности и степени подчиненности:*

- головные;
- дочерние;
- ассоциированные;
- филиалы.

Как видно из приведенной классификации, деятельность предприятия не обязательно связана с коммерческой деятельностью. Это может быть и государственная организация, и общественное, в том числе неформальное, объединение участников, связанных общей целью. Согласно более общему определению **предприятие** представляет собой комплексную систему культурных, технологических и процессных компонентов, организованных для достижения целей организации [1].

Интеграция отдельных компонентов в комплексную систему необходима для решения задач, которые не могут быть решены индивидуально. В «классической» книге Гради Буча [3] приведено описание структуры социальных институтов, к которым, несомненно, можно отнести и предприятия: «Одни организации быстро распадаются, другие функционируют на протяжении нескольких поколений. Чем больше организация, тем отчетливее проявляется в ней иерархическая структура. Транснациональные корпорации состоят из компаний, которые в свою очередь состоят из отделений, содержащих различные филиалы. Последним принадлежат уже отдельные офисы и т. д. Границы между частями организации могут изменяться, и с течением времени может возникнуть новая, более стабильная иерархия».

Предприятия (организации, фирмы) имеют право на добровольной основе объединять свою научно-техническую, производственную, коммерческую и другие виды деятельности, если это не противоречит действующему антимонопольному законодательству. Предприятия и другие первичные субъекты хозяйствования в рыночной экономике могут создавать разные по принципам и целям добровольные объединения:

• *ассоциации* — самая простая форма договорного объединения предприятий (фирм, компаний, организаций) с целью

постоянной координации хозяйственной деятельности. Ассоциация не имеет права вмешиваться в производственную и коммерческую деятельность любого из ее участников (членов);

- **картели** — договорные объединения предприятий преимущественно одной отрасли для осуществления совместной коммерческой деятельности и регулирования сбыта изготовленной продукции;

- **синдикаты** — организационная форма существования разновидности картельного соглашения, предусматривающего реализацию продукции участников через создаваемый совместный сбытовой орган или сбытовую сеть одного из участников объединения. Таким же образом может быть организована закупка сырья для всех участников синдиката. Эта форма объединения предприятий присуща отраслям с массовым производством однородной продукции;

- **тресты** — монополистические объединения предприятий, ранее принадлежавших различным предпринимателям, в единый производственно-хозяйственный комплекс. При этом предприятия полностью теряют свою юридическую и хозяйственную самостоятельность, поскольку интегрируются все направления их деятельности;

- **корпорации** — договорные объединения субъектов хозяйствования на основе интеграции их научно-технических, производственных и коммерческих интересов с делегированием отдельных полномочий для централизованного регулирования деятельности каждого из участников;

- **концерны** — форма уставных объединений предприятий, характеризующаяся единством собственности и контроля. Объединение происходит чаще всего по принципу диверсификации, когда один концерн интегрирует предприятия разных отраслей экономики (промышленности, транспорта, торговли, науки, банковского или страхового дела). После создания концерна субъекты хозяйствования теряют свою самостоятельность, подчиняясь мощным финансовым структурам. В современных условиях значительно расширяется сеть международных концернов;

- **консорциумы** — временные уставные объединения промышленного и банковского капитала для достижения общей цели (например, осуществления совместного крупного хозяйственного проекта). Участниками консорциума могут быть государственные и частные фирмы, а также отдельные страны (например, Международный консорциум спутниковой связи);

- **холдинги (холдинговые компании)** — специфическая организационная форма объединения капиталов: интегрированное общество, которое непосредственно не занимается производственной деятельностью, а использует свои средства для приобретения контрольных пакетов акций предприятий, являющихся участниками концерна или другого добровольного объединения, с целью контролирования деятельности таких предприятий. Объединяемые в холдинги субъекты имеют юридическую и хозяйственную самостоятельность, однако право решения основных вопросов их деятельности принадлежит холдинговой компании;

- **финансовые группы** — объединения юридически и экономически самостоятельных предприятий разных отраслей экономики. В отличие от концерна, во главе финансовых групп становятся один или два банка, которые распоряжаются капиталом предприятий, входящих в состав финансовых групп, координируют все сферы их деятельности.

В настоящее время основными формами добровольных объединений предприятий (фирм, организаций) все больше становятся концерны, корпорации и финансовые (промышленно-финансовые) группы.

Предприятия помимо внешней интеграции могут иметь сложную внутреннюю структуру, с которой также могут происходить изменения. На рис. 1.1. приведен пример производственной структуры машиностроительного предприятия [2].

В состав любого предприятия входят не только производственные подразделения, но и отделы аппарата управления, учреждения культурно-бытового назначения и др. Поэтому наряду с производственной структурой существует общая структура предприятия.

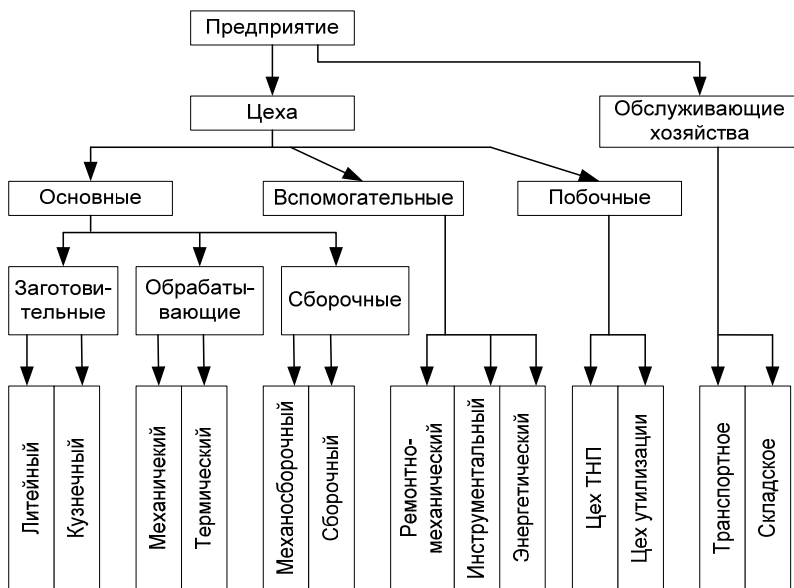


Рис. 1.1. Производственная структура машиностроительного предприятия

Общую структуру образует совокупность всех производственных, непроизводственных и управленческих подразделений предприятия. Типовая общая структура промышленного предприятия показана на рис. 1.2.

Предприятие возглавляет **директор**. Он осуществляет руководство предприятием в целом, т. е. представляет предприятие в любых организациях, распоряжается в пределах действующего законодательства его имуществом, заключает договора, открывает в банках расчетные счета и т. п.

Первым заместителем директора является **главный инженер**. Он руководит научно-исследовательскими и экспериментальными работами, непосредственно отвечает за совершенствование техники и технологии производства. В его обязанности входят также техническая подготовка и обслуживание производства, разработка мероприятий по повышению качества продукции и соблюдению технологической дисциплины.

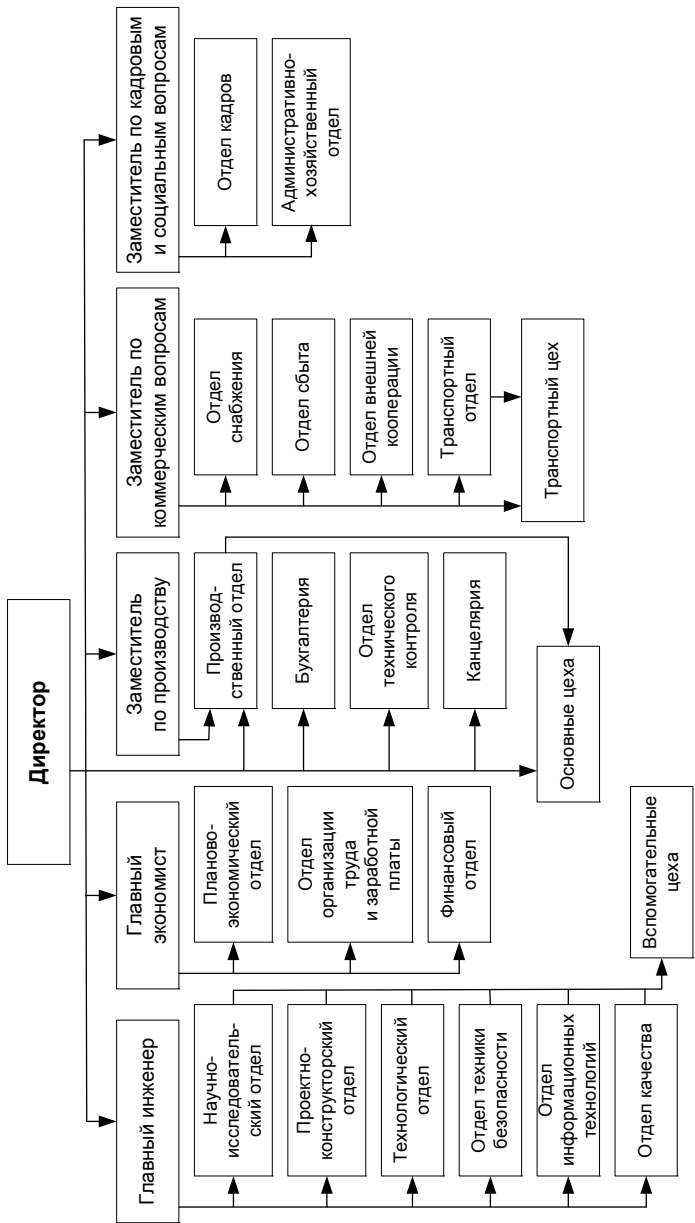


Рис. 1.2. Типовая общая структура промышленного предприятия

Экономическую службу на предприятии возглавляет **главный экономист** (заместитель директора по вопросам экономики), который отвечает за организацию плановой работы на предприятии. Подчиненные ему отделы осуществляют контроль над выполнением плановых задач, производят анализ деятельности предприятия. В его компетенции находятся также вопросы финансов, организации труда и заработной платы.

Главная задача **заместителя директора по производству** — обеспечение выполнения планов предприятия. С этой целью начальник производства и подчиненный ему производственный отдел, разрабатывающий оперативные планы выпуска продукции для каждого цеха, обеспечивают ритмичную работу по их выполнению, осуществляют контроль и регулирование производственного процесса.

Маркетинговые функции изучения спроса, рынков сбыта, рекламы, продвижения товаров, а также материально-технического обеспечения производства возложены на **заместителя директора по коммерческим вопросам**.

Заместитель директора по кадровым и социальным вопросам отвечает за реализацию кадровой политики предприятия, занимаясь вопросами отбора персонала, профессиональной ориентации и социальной адаптации, обучения, повышения в должности, увольнения и т. п. Кроме того, ему подчиняются службы, удовлетворяющие социальные потребности персонала.

Несколько подразделений аппарата управления предприятия подчинено непосредственно директору. Учет производства, контроль над использованием средств и соблюдением финансовой дисциплины, составлением баланса, расчеты с рабочими и служащими осуществляет **бухгалтерия**.

Функции контроля качества продукции, предупреждения брака, разработки и внедрения систем управления качеством возложены на независимый от любого заместителя директора **отдел технического контроля**.

Реализацию делопроизводства на предприятии, приемку входящей документации, ее регистрацию, учет, распределение,

организацию внешнего документооборота, отправку и хранение документов обеспечивает *канцелярия*.

Директору предприятия также непосредственно подчинены *начальники цехов*, осуществляющие техническое и хозяйственное руководство соответствующими подразделениями. В состав аппарата управления цехом входит заместитель начальника цеха по подготовке производства. Он занимается вопросами разработки технологических процессов, обеспечивает участки необходимой документацией и оснасткой. Помощник начальника цеха по производству осуществляет оперативное руководство производственными процессами. Механик цеха организует ремонт оборудования и надзор за его эксплуатацией. Начальник цеха осуществляет управление производством с помощью *мастеров производственных участков*, которые уже непосредственно или через бригадиров организуют труд исполнителей. Экономической работой в пределах цеха занимается *экономист*, а вопросами нормирования и оплаты труда — *нормировщик*.

1.1.2. Понятие архитектуры предприятия

На фоне вышеприведенных определений, классификаций и структур предприятий становится явной принципиальная значимость изучения дисциплины «Архитектура предприятия (организации)». В общем виде под **архитектурой организации** понимается всестороннее и исчерпывающее описание (модель) всех ее ключевых элементов и межэлементных отношений [4]. Похожее, но более подробное определение приведено в свободной энциклопедии Википедия [5]: «**Архитектура предприятия** — это наиболее общее и всестороннее представление предприятия как хозяйствующего субъекта, имеющего краткосрочные и долгосрочные цели ведения своей основной деятельности, определенные миссией на региональном и мировом рынке и стратегией развития, внешние и внутренние ресурсы, необходимые для выполнения миссии и достижения поставленных целей, а также сложившиеся правила ведения основной деятельности (бизнеса)».

Согласно стандарту по формальному описанию архитектуры предприятия ISO 15704 [6], предложенному рабочей группой IFAC/IFIP (International Federation of Automatic Control / International Federation for Information Processing), **архитектура организации** должна включать роль людей, описание процессов (функции и поведение) и представление всех вспомогательных технологий на протяжении всего жизненного цикла организации.

В соответствии с [7] **архитектура организации** является стратегической информационной основой, определяющей следующие компоненты:

- 1) структуру бизнеса;
- 2) информацию, необходимую для ведения бизнеса;
- 3) технологии, применяемые для поддержания бизнес-операций;
- 4) процессы преобразования, развития и перехода, необходимые для реализации новых технологий в ответ на изменение/появление новых бизнес-потребностей.

Компанией Gartner, являющейся лидером среди разработчиков архитектуры предприятия, даются следующие определения [8]:

1) **архитектура предприятия** — это общий план или концепция, используемая для создания системы, такой как здание или информационная система, или «абстрактное описание системы, ее структуры, компонентов и их взаимосвязей»;

2) **архитектура предприятия** — это семейство руководящих принципов, концепций, правил, шаблонов, интерфейсов и стандартов, используемых при построении совокупности информационных технологий (ИТ) предприятия.

Первое определение сфокусировано на описании существующих и будущих систем, второе — на процессе их построения.

В курсе лекций Данилина А.В. и Слюсаренко А.И. [1] вводится иерархический принцип определения архитектуры предприятия (рис. 1.3), суть которого состоит в следующем:

- **архитектура предприятия** определяет общую структуру и функции систем в рамках всей организации в целом и обеспечивает общую рамочную модель, стандарты и руководства для архитектуры уровня отдельных проектов. Общее видение создает

возможность единого проектирования систем, адекватных с точки зрения удовлетворения потребностей организации и способных к взаимодействию и интеграции там, где это необходимо;

- **архитектура уровня отдельных проектов** определяет структуру и функции систем на уровне проектов и программ, но в контексте организации в целом, т. е. не индивидуальных изолированных систем;

- **архитектура прикладных систем** определяет структуру и функции приложений, которые разрабатываются с целью обеспечения требуемой функциональности. Некоторые элементы этой архитектуры могут быть определены на уровне архитектуры предприятия или архитектуры отдельных проектов (в форме стандартов и руководств) с целью использования лучшего опыта и соответствия принципам всей архитектуры в целом.

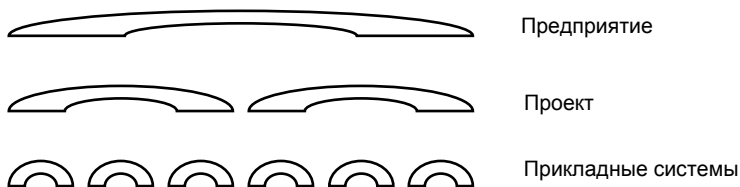


Рис.1.3. Уровни архитектуры предприятия

Стандарт IEEE 1471 [9], разработанный Institute of Electrical and Electronics Engineers¹, предоставляет метамодель для определения архитектуры предприятия (рис. 1.4) — **Conceptual Framework**. Стандарт определяет такие абстрактные элементы архитектуры, как представления, системы, среды, обоснования, заинтересованные стороны и т. д. В соответствии со стандартом система обладает некоторой архитектурой, которая может быть определенным образом описана с различных точек зрения в зависимости от интереса участников процесса анализа и построения

¹ Институт инженеров-электриков и электронщиков — международная организация, занимающаяся в частности изданием рекомендаций (международных стандартов). Член организаций ANSI и ISO.

архитектуры системы. Каждой точке зрения соответствует определенное представление, основу которого составляет некоторый набор моделей. Однако стандарт IEEE 1471 не определяет структуру собственно архитектуры предприятия. Например, в стандарте указывается на необходимость иметь различные представления архитектуры, но при этом не указывается, какие именно.

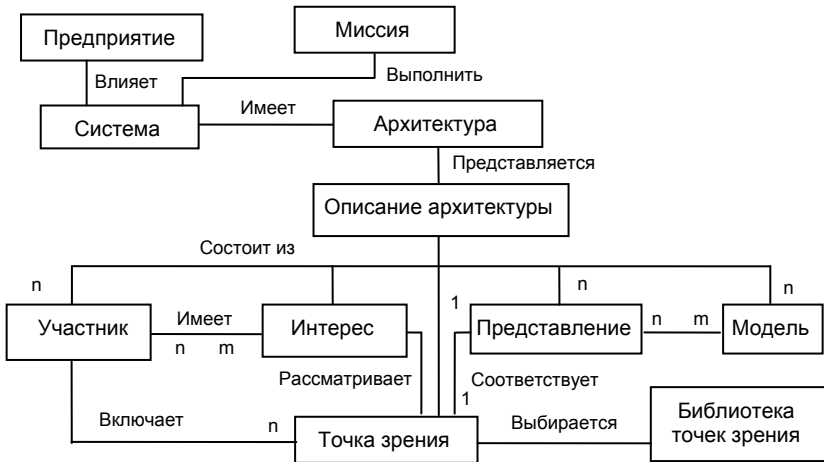


Рис. 1.4. Метамодел для определения архитектуры предприятия стандарта IEEE 1471

С учетом вышеприведенных определений автором предлагается обобщенное понятие **архитектуры предприятия** как комплексного представления предприятия в статическом и динамическом аспектах. *Статический аспект* отражает характеристику предприятия в некоторый фиксированный момент времени со стороны трех основных компонентов: миссии, бизнес-архитектуры и системной архитектуры. *Динамический аспект* представляет собой описание процесса перехода предприятия от текущего состояния к некоторому желаемому состоянию в будущем.

1.1.3. Значение архитектуры предприятия в современных условиях

В современных условиях возникает необходимость изыскания возможностей эффективного использования существующих технологий организации бизнес-процессов предприятия и внедрения новых, что может быть обеспечено в рамках построения архитектуры предприятия. Таким образом, построение архитектуры предприятия является одним из главных средств управления изменениями, направленными на реализацию следующих возможностей:

- оказание помощи менеджерам при анализе потенциальных изменений и их реализации;
- предоставление основы для совместной работы бизнес-менеджеров и ИТ-менеджеров над целями, бизнес-процессами и системной организацией;
- предоставление единого хранилища всей информации о предприятии;
- обеспечение менеджерам поддержки в принятии решений.

Сутью концепции корпоративной архитектуры предприятия является разработка плана использования ИТ-ресурсов в бизнес-процессах и совокупности принципов управления, отражающих стратегию бизнеса через информационные технологии. Хотя в концепции и не описываются конкретные технические решения для отдельных информационных систем, ее использование позволяет получить значительную выгоду для бизнеса предприятия в целом, что выражается в повышении эффективности эксплуатации информационных систем, снижении рисков инвестиций в ИТ, повышении гибкости технологических решений и возможности относительно простой адаптации под изменяющиеся внешние условия и требования бизнеса.

Построение эффективной архитектуры позволяет предприятию снизить риски и увеличить отдачу от инвестиций в информационные технологии, что достигается посредством четкого определения структуры существующих и вновь проектируемых

автоматизированных информационных систем. Наличие обоснованных стратегий позволяет упростить и ускорить выполнение бизнес-процессов посредством проведения их реинжиниринга во взаимосвязи с используемыми ИТ.

Итак, имеется **три причины, обуславливающие необходимость использования архитектурного подхода** [4]:

1) рост масштаба и сложности информационных технологий, увеличение их стоимости и повышение степени риска в проектах их создания и внедрения;

2) включение ИТ в основную деятельность, рост требований к эффективности инвестиций в ИТ;

3) переход к процессному подходу, интегрирующему деятельность подразделений, рост требований к эффективному взаимодействию ИТ-систем между собой.

В результате использования архитектурного подхода обеспечивается [4]:

- информационная поддержка работ по сопровождению и развитию ИТ-инфраструктуры, которая включает:

- выявление бизнес-процессов, требующих первоочередной автоматизации;

- выявление первоочередных направлений совершенствования каналов связи;

- анализ ИТ-систем и их взаимодействия, оценку степени покрытия бизнес-процессов и информационных потоков существующими системами;

- оптимизацию обработки информации во взаимодействующих системах (избавление от дублирующих систем и данных, согласование справочников и классификаторов, используемых в различных системах, и т. п.);

- выявление, согласование, формализацию и документирование требований к перспективным ИТ-системам, контроль внедрения новых систем на предмет соответствия согласованным требованиям в части покрытия информационных потоков;

- анализ альтернативных вариантов совершенствования ИТ-инфраструктуры;

- информационная поддержка работ по совершенствованию бизнес-процессов предприятия, позволяющая осуществлять:
 - выявление бизнес-процессов, требующих совершенствования;
 - избавление от дублирующих действий в различных системах;
 - анализ альтернативных вариантов совершенствования бизнес-процессов;
- информационная поддержка всех заинтересованных лиц, включая сотрудников предприятия, использующих ИТ-системы в силу своих должностных обязанностей, а также разработчиков и лиц, сопровождающих используемые на предприятии системы. При этом все заинтересованные лица обеспечиваются единым языком базовых представлений.

1.2. Статический и динамический аспекты архитектуры предприятия

1.2.1. Основные элементы и слои архитектуры предприятия

Рассматриваемая в статике архитектура предприятия состоит из следующих элементов [10]:

- 1) миссии и стратегии, стратегических целей и задач;
- 2) бизнес-архитектуры;
- 3) системной архитектуры.

Миссия — одно из основополагающих понятий стратегического управления. Разные ученые давали различные формулировки миссии, но в целом можно выделить два подхода к пониманию миссии [11]:

1) **в широком смысле миссия** — это философия и предназначение, смысл существования организации. *Философия организации* определяет ценности и принципы, в соответствии с которыми организация намеревается осуществлять свою деятельность.

Предназначение определяет действия, которые организация намерена осуществлять, и цели, которых она намерена достичь. Философия организации определяется на этапе становления организации и редко меняется, тогда как предназначение может меняться в процессе изменений как внутренней, так и внешней среды организации;

2) **в узком понимании миссия** — это определение целей и причин существования организации, т. е. миссия в таком понимании должна раскрывать смысл существования организации, в котором проявляется отличие данной организации от ей подобных.

Стратегия в широком толковании представляет собой искусство руководства общественными процессами, общий план руководства. Применительно к предприятию (организации), по определению Е. Кассельса [5], **стратегия** — это модель поведения, которой следует организация для достижения своих долгосрочных целей. В свою очередь, **стратегическое планирование** — это процесс осуществления систематизированных и взаимосогласованных работ с определением долгосрочных целей и направлений деятельности предприятия [12].

Четкое определение миссии, стратегии и бизнес-целей предприятия позволяет сформулировать основные направления его развития и поставить долгосрочные цели и задачи.

Бизнес-архитектура на основе миссии, стратегии развития и долгосрочных бизнес-целей определяет необходимые организационную структуру и функциональную модель предприятия, описывающую направленные на реализацию текущих задач и перспективных целей бизнес-процессы. Бизнес-архитектура является областью деятельности высших руководителей, отвечающих за основные функции (бизнес) организации, и, как правило, содержит утверждения по поводу миссии и целей организации, критических факторов успеха, бизнес-стратегии, описания функций, а также структур и процессов, необходимых для их реализации.

Системная архитектура (ИТ-архитектура, архитектура ИС) определяет совокупность технологических и технических решений для обеспечения информационной поддержки работы

предприятия в соответствии с правилами и концепциями, определенными бизнес-архитектурой.

Подробное описание основных элементов архитектуры предприятия приведено в пп. 1.2.2–1.2.4.

Представление архитектуры предприятия в статическом аспекте как совокупности трех основных элементов определяет позиционирование слоев архитектуры.

В архитектуре предприятия выделяют следующие слои:

- 1) фронт-офис (Front-Office);
- 2) мидл-офис (Middle-office);
- 3) бэк-офис (Back-office);
- 4) учет (Accounting);
- 5) информационное хранилище (Data Warehouse);
- 6) отчетность (Reporting).

Первые четыре присутствуют как в бизнес-архитектуре, так и в системной архитектуре. Два последних слоя относятся только к системной архитектуре (табл. 1.1).

Таблица 1.1

Наличие слоев в архитектуре предприятия

Слои архитектуры предприятия	Бизнес-архитектура	Системная архитектура
Фронт-офис (Front-Office)	Да	Да
Мидл-офис (Middle-office)	Да	Да
Бэк-офис (Back-office)	Да	Да
Учёт (Accounting)	Да	Да
Информационное хранилище (Data Warehouse)	Нет	Да
Отчётность (Reporting)	Нет	Да

Фронт-офис в бизнес-архитектуре — это совокупность бизнес-процессов, процедур, нормативных документов (регламентов), справочников, печатных форм, организационно-штатных подразделений, обеспечивающих со стороны предприятия взаимодействие с клиентом:

- получение и ввод для последующей обработки первичных документов;

- печать и предоставление клиенту информации и документов;
- рассылку клиентам информационных сообщений;
- «обзвон» клиентов;
- прием входящих телефонных звонков клиентов;
- прием запросов и предоставление информации.

Примеры подразделений фронт-офиса: Call-центр, подразделение операционного обслуживания, касса (отдельные бизнес-процессы).

Фронт-офис в системной архитектуре — это совокупность информационных систем, баз данных (БД) и справочников, направленных на автоматизацию бизнес-процессов взаимодействия с клиентом. Примеры информационных систем фронт-офиса: интернет-банк, информационная система Call-центра, система управления взаимоотношениями с клиентами (CRM — Customer Relationship Management System).

Мидл-офис в бизнес-архитектуре — это совокупность бизнес-процессов, процедур, нормативных документов (регламентов), справочников, печатных форм, организационно-штатных подразделений, обеспечивающих подготовку и принятие решений. Примеры подразделений мидл-офиса: подразделение проверки заемщиков в службе безопасности, подразделение управления рисками.

Мидл-офис в системной архитектуре — это совокупность информационных систем, баз данных и справочников, направленных на автоматизацию бизнес-процессов, связанных с подготовкой и принятием решений. Примеры информационных систем мидл-офиса: система ведения позиционного учета, система проверки заемщика в бюро кредитных историй, система расчета скорингового¹ балла по кредитной заявке.

¹ Скоринговый балл — количественная оценка кредитоспособности потенциального заемщика (чем выше балл, тем выше кредитоспособность). В зависимости от балла определяется либо вероятность дефолта заемщика, либо принадлежность к определенному классу.

Бэк-офис в бизнес-архитектуре — это совокупность бизнес-процессов, процедур, нормативных документов (регламентов), справочников, печатных форм, организационно-штатных подразделений, реализующих журнальный (регистрационный) учет операций, совершенных клиентом. Как правило, регистрационный учет представляет собой журнал операций клиентов. Этот учет не связан с бухгалтерскими счетами, не является двухсторонним. Примером подразделения бэк-офиса является подразделение розничного кредитования.

Бэк-офис в системной архитектуре — это совокупность информационных систем, баз данных и справочников, реализующих журнальный (регистрационный) учет операций, совершенных клиентом. В современной системной архитектуре крупных корпоративных информационных систем данный класс систем представлен недостаточно широко. К данному классу относится большинство систем учета финансов для личного использования.

Учет в бизнес-архитектуре — это совокупность бизнес-процессов, процедур, нормативных документов (регламентов), справочников, печатных форм, организационно-штатных подразделений, бизнес-процессов, реализующих ведение бухгалтерского учета и отчетности по Российским правилам бухгалтерского учета (РПБУ) и Международным стандартам финансовой отчетности (МСФО), ведение баланса предприятия.

На данном уровне часто реализован также налоговый учет. Формирование проводок бухгалтерского учета происходит на основании журнала операций бэк-офиса путем разнесения операций в соответствии со справочником котировок.

Учет в системной архитектуре — это совокупность информационных систем, баз данных и справочников, реализующих ведение бухгалтерского учета и отчетности по РПБУ и МСФО, ведение баланса предприятия. Данный класс систем часто реализует также налоговый учет. Формирование проводок бухгалтерского учета происходит на основании журнала операций бэк-офиса путем разнесения операций в соответствии со справочником котировок.

Информационное хранилище в системной архитектуре представляет собой совокупность информационных систем, баз данных и справочников, реализующих функциональность по описанию метаданных, сбору, очистке, обогащению, консолидации первичной информации из транзакционных систем, а также построению витрин данных.

Отчётность в системной архитектуре — это совокупность информационных систем, баз данных и справочников, автоматизирующая построение отчётности на основе данных из информационного хранилища. Примеры систем отчётности: система управленческой отчётности, система аналитической отчётности, система ключевых показателей эффективности подразделений предприятия, система формирования показателей для расчёта скорингового балла по кредитной заявке.

В динамике архитектура предприятия представляет собой логически связанный цельный план действий и скоординированных проектов, необходимых для преобразования сложившейся архитектуры организации к состоянию, определенному как долгосрочная цель, базирующийся на текущих и планируемых бизнес-целях и бизнес-процессах организации [10].

Таким образом, архитектура предприятия в общем случае описывается следующими последовательно зависимыми разделами:

- сформулированными миссией и стратегией предприятия, стратегическими целями и задачами;
- бизнес-архитектурой в текущем (*AS-IS — как есть*) и планируемом (*TO-BE — как должно быть*) состоянии;
- системной архитектурой в текущем (AS-IS) и планируемом (TO-BE) состоянии;
- планами мероприятий и проектов по переходу из текущего состояния в планируемое (планами миграции). Иногда модели AS-IS и TO-BE различаются очень сильно, так что переход от начального к конечному состоянию становится неочевидным. В этом случае необходима третья модель, описывающая процесс перехода от начального состояния системы к конечному, поскольку такой переход — это тоже бизнес-процесс.

Планы миграции определяют сценарий перехода предприятия от текущего состояния к перспективному, определяемому стратегическими целями и задачами, а также преобразования как бизнес-архитектуры, так и системной архитектуры. При поэтапной миграции для целей формализации промежуточных результатов разрабатываются один или несколько промежуточных (миграционных) указанных элементов архитектуры. Планы миграции в соответствии с принятой на предприятии методологией управления проектами формализуются в виде отдельных проектов, включающих, в частности:

- определение проекта как совокупности задач и работ;
- фазы и сроки реализации проекта в целом и составляющих проект задач и работ;
- анализ конкурентной среды и рисков, связанных с реализацией проекта;
- состав статей расхода бюджета проекта;
- критерии успешности реализации проекта и ожидаемый экономический эффект.

Выполнение плана миграции не означает замораживания развития бизнес- и системной архитектуры [10] (рис. 1.5).

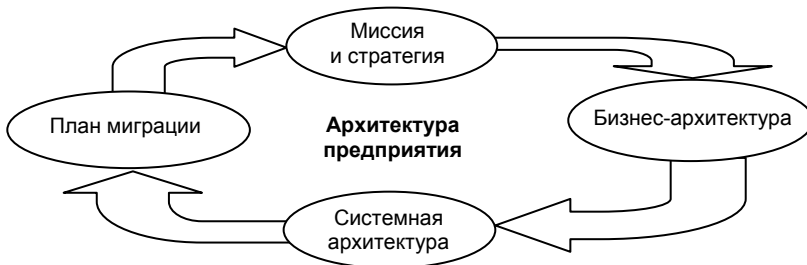


Рис. 1.5. Циклическое развитие архитектуры предприятия

Таким образом, планируемая системная архитектура является архитектурой TO-BE только на определенном витке развития предприятия. Одновременно возврат к стратегическому уровню миссии и стратегических целей и задач не означает необходи-

мость пересмотра миссии и стратегии. Но в конце каждого цикла обязательно проводится анализ эффективности разработанных и осуществленных мероприятий, при необходимости при второй итерации корректируются бизнес-архитектура, системная архитектура, реализуются новые планы миграции. В каждый момент времени может быть несколько циклов, каждый такой цикл не обязательно затрагивает все предприятие в целом, цикл может затрагивать отдельные направления, отдельные вопросы бизнеса и может быть зафиксирован в виде отдельного проекта.

Технология проектирования информационных систем подразумевает сначала создание модели AS-IS, затем анализ и улучшение бизнес-процессов, т. е. создание модели TO-BE, и только на основе модели TO-BE строится модель данных, прототип и затем окончательный вариант информационной системы.

Связь понятий бизнес-архитектуры, системной архитектуры и архитектуры предприятия показана на рис. 1.6 [1].



Рис. 1.6. Позicionирование понятия «архитектура предприятия»

Эволюция представлений об архитектуре предприятия и получаемая на каждом этапе расширения соответствующих представлений дополнительная ценность показаны на рис. 1.7 [1].

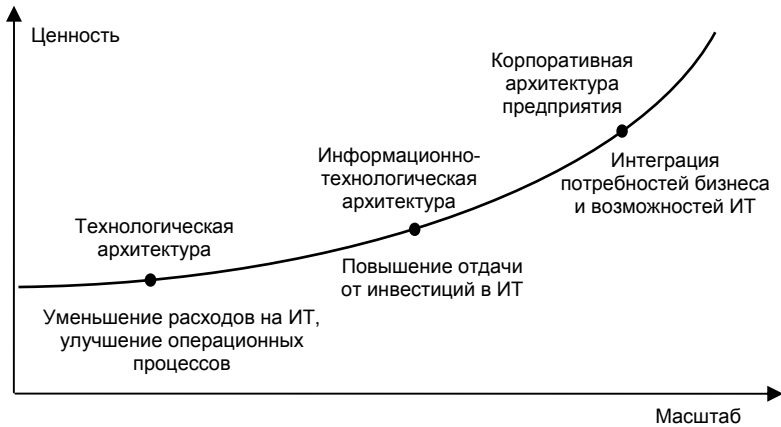


Рис. 1.7. Расширение представлений об архитектуре предприятия и дополнительные преимущества

Концепция архитектуры предприятия явилась результатом поиска некоторого целостного подхода, обеспечивающего «взгляд на организацию в целом» с учетом всех возможных измерений, хотя такой учет предполагает и усложнение представлений об архитектуре.

1.2.2. Миссия и стратегическое планирование

Миссия (см. п. 1.2.1) формулируется, как правило, в двух вариантах:

1) **короткий вариант миссии** представляет собой «брендовый слоган» организации (1–2 коротких предложения), направленный, прежде всего, на формирование имиджа организации в обществе;

2) **расширенный вариант миссии** предназначен для внутреннего пользования и должен подробно раскрывать все необходимые аспекты миссии:

- цель функционирования организации;
- область деятельности организации;

- философию организации;
- методы достижения поставленных целей;
- методы взаимодействия организации с обществом (социальную политику организации).

Правильно определенная миссия хотя и имеет всегда общий философский смысл, тем не менее обязательно несет в себе что-то, что делает ее уникальной в своем роде, характеризующей именно ту организацию, в которой она была выработана. На основе общей миссии предприятия формулируются и устанавливаются общефирменные цели, которые должны отвечать определенным требованиям, суть которых состоит в следующем:

1) **конкретность и возможность измерения.** Формулирование целей в конкретных формах создает исходную базу для последующих правильных хозяйственных и социальных решений. Благодаря этому можно более точно определить, насколько эффективны действия предприятия по осуществлению поставленных целей;

2) **ориентированность во времени** (обозначение конкретных горизонтов прогнозирования). Цели устанавливаются на длительные или короткие промежутки времени. Долгосрочная цель имеет горизонт прогнозирования равный нескольким годам (пять, семь, десять лет); краткосрочная — в пределах одного года;

3) **достижимость и направленность на повышение эффективности деятельности предприятия.** Недостижимые или достижимые лишь частично цели приводят к негативным последствиям: блокированию стремления работников эффективно хозяйствовать, снижению уровня их мотивации, ухудшению показателей инновационной, производственной и социальной деятельности предприятия, снижению конкурентоспособности его продукции на рынке;

4) **взаимосогласованность и поддерживаемость множественных целей предприятия**, т. е. действия и решения, необходимые для достижения одной цели, не могут препятствовать реализации других целей. Иное может привести к возникновению конфликтной ситуации между подразделениями предприятия (фирмы), ответственными за достижение различных целей.

Стратегия как модель организации бизнес-процессов предприятия для достижения долгосрочных целей формулирует общие направления развития деятельности предприятия, в первую очередь касающиеся производимой продукции и каналов ее продвижения. При этом стратегия должна обеспечить концентрацию усилий в той области, где будут иметь место устойчивые конкурентные преимущества. Разработка корпоративной стратегии позволяет перейти от управления организацией, зависящего от воздействия случайно возникающих внешних и внутренних факторов, к планомерной деятельности по достижению определенных результатов с возможностью оценки их достижимости по определенным критериям и применения адекватных управляющих воздействий [4]. На рис. 1.8 [2] отражены главные направления деятельности предприятия, которые могут декларироваться стратегией предприятия.

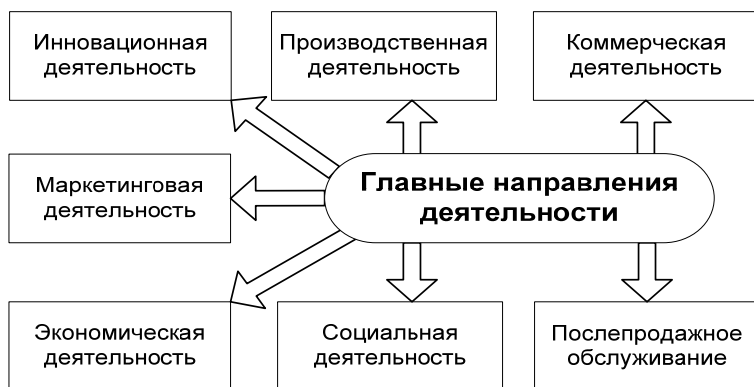


Рис. 1.8. Главные взаимосогласованные направления деятельности предприятия

По своему содержанию стратегия является долгосрочным плановым документом, результатом стратегического планирования (см. п. 1.2.1). **Основные этапы стратегического планирования** показаны на рис. 1.9.

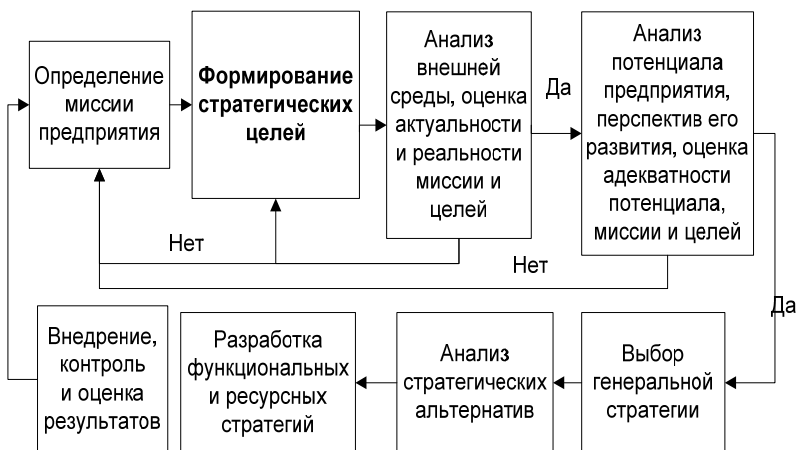


Рис. 1.9. Основные этапы стратегического планирования

Исходя из общей миссии предприятия формируются его стратегические цели. Реальность и эффективность стратегии предприятия обеспечиваются при условии формулирования стратегических целей с учетом следующих требований [12]:

- конкретности формулировки целей и их измеряемости;
- четкой ориентированности целей во времени;
- достижимости, сбалансированности и ресурсной обеспеченности;
- однонаправленности и взаимосоответствия целей.

При этом желательно устанавливать цели для каждого направления деятельности предприятия.

После определения миссии и целей начинается *диагностический этап* стратегического планирования. Первым важным шагом диагностического этапа является изучение внешней среды. Анализ внешней среды представляет собой непрерывный процесс наблюдения, изучения и контроля внешних для предприятия факторов с целью своевременного исчерпывающего определения возможных положительных явлений либо угроз субъекту хозяйствования.

Для разработки и осуществления стратегии большое значение имеет *анализ рыночных факторов*, проводимый в рамках анализа внешней среды. Рыночные факторы вследствие своей постоянной и высокой изменчивости оказывают непосредственное влияние на успешность предприятия. Речь идет в первую очередь о микроэкономическом анализе спроса, предложения и уровня конкуренции по определенной системе показателей [12].

Методы выбора генеральной стратегии можно разделить на две группы (рис. 1.10):

- 1) методы однопродуктового анализа;
- 2) методы «портфельного» анализа (матричные).

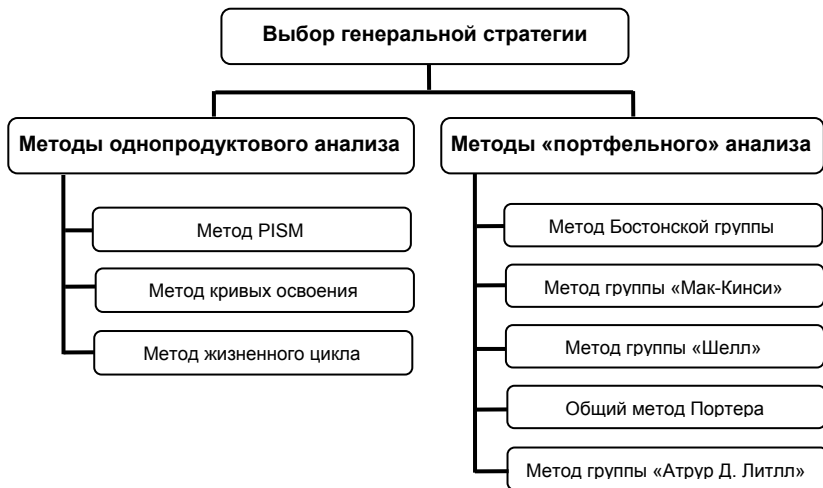


Рис. 1.10. Возможные методы выбора генеральной стратегии предприятия

Среди **методов однопродуктового анализа** наиболее научно обоснованным является *метод PIMS* (Profit Imprakt of Marketing Strategy), впервые реализованный компанией «General electric» с участием гарвардской школы бизнеса в начале 70-х гг. XX в. В основу метода положено моделирование влияния стратегических факторов на показатели эффективности предприятия (в частности, рентабельность капитальных вложений, валовую прибыль).

Если метод PIMS и его модели учитывают действие как внешних, так и внутренних факторов, то *метод кривых освоения*, который строится на зависимости размеров затрат на производство от его объема, отражает влияние внутренних факторов. Основой метода служит известная закономерность: рост масштабов производства обеспечивает экономию определенных затрат, размер которых не зависит или мало зависит от изменения количества единиц продукции. К тому же в процессе производства имеет место повторение операций, формирование навыков или динамического стереотипа, что также ведет к уменьшению трудозатрат.

Логически связанным с методом кривых освоения является еще один метод разработки стратегии предприятия — *метод жизненного цикла изделия*. За период своего существования изделие проходит, как правило, четыре стадии: внедрение (освоение), рост, зрелость, спад. Принятию стратегического решения по конкретному изделию предшествует идентификация стадии его жизненного цикла. В процессе идентификации одну стадию отличают от другой с помощью таких показателей, как рост объемов продажи (производства), количество конкурентов, темпы технологических изменений, частота модификации изделия и т. п.

Для каждой стадии определяются приоритетные стратегические направления и действия. В частности, на стадии роста таким направлением является маркетинговая деятельность (наступательная реклама и активное товаропродвижение, оптимизация распределения товара, улучшение ценообразования, адекватная реакция на спрос продукции и т. п.). На стадии зрелости на первый план выдвигаются показатели эффективности производства и коммерческой деятельности (оптимальное использование производственного потенциала, стандартизация коммерческих процедур, постепенное уменьшение затрат на исследовательские работы по данному изделию). С целью наиболее полного учета действий внешних факторов строится матрица, причем одним из образующих показателей является характеристика конкурентной позиции данного изделия.

Почти все методы выбора стратегии предприятия в условиях диверсифицированного производства являются **матричными**,

в том числе и **методы портфельного анализа**. Использование этих методов происходит по одинаковой схеме: как правило, строится матрица, на одной оси которой размещаются оценки перспектив развития рынка, на другой — оценка конкурентоспособности так называемого стратегического центра хозяйствования (СЦХ). Определяются миссия и цели каждого такого центра, генеральная стратегия и ее подстратегии. Стратегические планы каждого центра оцениваются центральным аппаратом управления, после чего определяются основные показатели стратегии предприятия в целом.

Главное отличие разных матричных методов состоит в различных показателях, используемых для оценки привлекательности рынка и конкурентной позиции стратегических центров. Наиболее простым и самым распространенным является **метод Бостонской консультативной группы** (БКГ). Показателями, формирующими оценочную матрицу по этому методу, являются темп роста и контролируемая данным предприятием часть рынка.

Среди матричных методов известен также **метод консультационной группы «МакКинси»**, где оценочными показателями являются конкурентная позиция стратегических центров (слабая, средняя, сильная) и привлекательность рынка (аналогичные три оценки). При помощи данного метода изучают и анализируют специфическое действие на каждом рынке определенной совокупности факторов, включающей емкость и темпы роста рынка, динамику уровня цен, контролируемую предприятием часть рынка, цикличность спроса, тенденции изменения количества конкурентов, конкуренцию, преимущества лидеров отрасли, темпы роста прибыли лидеров, состояние трудовых ресурсов.

Из других матричных методов определения генеральных стратегий известны следующие:

- **общий стратегический метод Портера** (стратегические преимущества/стратегические цели)¹;

¹ При перечислении матричных методов в скобках указываются показатели, которые формируют матрицу.

- *метод консультационной группы «Артур Д. Литлл»* (стадия жизненного цикла/конкурентная позиция);

- *метод консультационной группы «Шелл»* (потенциальный рынок/мощность предприятия).

Стержнем стратегического плана предприятия является **базовая стратегия**. В соответствии с циклом развития предприятия можно выбрать одну из базовых стратегий:

- стратегию роста, характеризующую намерение предприятия увеличивать объемы продажи, прибыли, капитальных вложений и т. п.;

- стратегию стабилизации при ощутимой нестабильности объемов продажи и прибыли;

- стратегию выживания — сугубо оборонную стратегию, применяемую при глубоком кризисе предприятия.

В рамках соответствующей базовой стратегии можно выбрать одно из нескольких возможных действий, которые принято называть стратегическими альтернативами (табл. 1.2).

По функциональному признаку базовая стратегия может быть представлена маркетинговой стратегией, стратегией научно-исследовательских и экспериментально-внедренческих работ, производственной стратегией.

По ресурсному признаку выделяют стратегии кадров и социального развития, стратегии материально-технического обеспечения, организационные стратегии, инвестиционные стратегии, стратегии технического развития и финансовые стратегии.

Каждая базовая стратегия, как правило, содержит следующие компоненты:

- 1) цели, условия и основные направления деятельности в той или иной сфере, конечные результаты, обеспечивающие внедрение ресурсных стратегий;

- 2) порядок и последовательность (в пространстве и во времени) решения качественных и количественных задач долгосрочных планов; ряд мероприятий, адекватных назначению базовой стратегии, что обеспечивает достижение поставленной цели.

Таблица 1.2

Стратегические альтернативы на основе базовой стратегии

Варианты базовой стратегии	Критерии стратегии	Стратегические альтернативы
1. Стратегия роста	<ol style="list-style-type: none"> 1. Объем продаж. 2. Доход. 3. Доля рынка. 4. Скорость рынка¹ 	<ol style="list-style-type: none"> 1. Интенсификация рынка: захват новых рынков; географическая экспансия. 2. Диверсификация. 3. Межфирменное сотрудничество и кооперация. 4. Внешнеэкономическая деятельность
2. Стратегия стабилизации (наступательно-оборонительная)	<ol style="list-style-type: none"> 1. Доход от объема продаж. 2. Доход от активов 3. Доход от акций. 4. Скорость оживления деятельности² 	<ol style="list-style-type: none"> 1. Экономия: ревизия затрат, консультация, оживление деятельности. 2. Сдвиги: уменьшение затрат, восстановление дохода, активизация финансовой деятельности. 3. Обеспечение устойчивости: селективность, балансирование на рынках, финансовая экономия
3. Стратегия выживания (оборонительная)	<ol style="list-style-type: none"> 1. Показатели анализа продуктов и рынков. 2. Показатели анализа финансового состояния. 3. Показатели анализа управления 	<ol style="list-style-type: none"> 1. Перестройка маркетинговой деятельности: изъятие товара из продажи

¹ Изменение цены во времени.

² Восстановление либо увеличение экономических показателей деятельности предприятия во времени.

Компоненты стратегического планирования отражаются в бизнес-плане предприятия. **Бизнес-план предприятия** — это документ, в котором изложены сущность, направления и способы реализации предпринимательской идеи, охарактеризованы рыночные, производственные, организационные и финансовые аспекты будущего бизнеса, а также особенности управления. Этот документ является разрешительным основанием привлечения инвестиций для разработки и реализации предпринимательской идеи в виде инновационно-инвестиционных проектов.

Помимо стратегического планирования выделяют тактическое и оперативное планирование.

1.2.3. Бизнес-архитектура

Бизнес-архитектура как составной элемент архитектуры предприятия (см. п. 1.2.1) на основании миссии, стратегии развития и долгосрочных бизнес-целей определяет необходимые *организационную структуру* и *бизнес-процессы*, описываемые функциональной моделью предприятия и используемые в процессе разработки и реализации продуктов.

Бизнес-архитектура включает [10]:

- предлагаемые и планируемые к реализации продукты и услуги (включая индивидуальные схемы их производства), формализованные в виде единого реестра продуктов и услуг;
- каналы продажи продуктов и услуг, построенные как на базе структурных и территориальных подразделений предприятия, так и на базе современных информационных технологий (ИТ);
- функции и процессы по реализации внешних и внутренних продуктов и услуг, образующие деревья функций и процессов (бизнес-функции и бизнес-процессы);
- финансовые и распорядительные документы (как в бумажном, так и в электронном виде), формализованные в виде единого реестра (альбома форм) документов предприятия;
- документопотоки, определяемые нормативными актами по внутреннему и внешнему документообороту;

- организационную структуру предприятия, включающую штатное расписание предприятий и его территориальных подразделений, являющихся самостоятельными хозяйствующими единицами (юридическими лицами), комитеты, рабочие группы и ролевые функции отдельных сотрудников, должностные инструкции, положения о подразделениях и рабочих органах и другие документы, регламентирующие взаимоотношения и распределение ответственности между сотрудниками, а также между структурными подразделениями.

В теории управления **организационная структура** определяется как абстрактная категория, характеризующаяся тремя параметрами: степенью сложности, степенью формализации и степенью централизации.

Сложность отражается множеством отличительных признаков организации. Чем глубже разделение труда, чем больше вертикальных уровней в иерархии управления и структурных подразделений, тем сложнее координировать деятельность людей в организации. Объем разработанных правил и процедур, руководствуясь которыми организация направляет поведение своих сотрудников, и представляет собой *формализацию*. Чем больше правил и регуляторов, определяющих объем запрещенных и разрешенных действий работников организации, тем более формализованной является структура организации. *Централизация* определяет место, где сосредоточено право принятия решений. Если все решения (или их большинство) принимаются высшими руководителями, тогда организация является централизованной. Децентрализация означает, что право принятия решений передается (делегировается) с высших уровней управления на низшие уровни.

В процессе реализации вышеназванных функций задача менеджмента заключается в разработке для всех компонентов организации такой формы и способов «состыковки» друг с другом, при которой предприятие не утрачивает целостности и функционирует максимально эффективно. Поэтому на любом предприятии имеет место соподчиненность его составных частей, уровней управления, четкое разделение власти, прав и ответственности.

При формировании организационной структуры предприятия необходимо руководствоваться несколькими правилами [11]:

1) стратегически важные для компании виды деятельности должны стать основными звеньями структуры, а статус их руководителей должен соответствовать значимости видов деятельности;

2) при изменении стратегии обязательна корректировка организационной структуры;

3) сторонним организациям можно передавать виды деятельности, которые выполняются ими дешевле, быстрее и более качественно. Как правило, эти виды деятельности не являются основными для компании, но иногда передаются и основные, если последние не определяют конкурентное преимущество компании;

4) все аспекты стратегически значимых видов деятельности должны быть подотчетны одному руководителю;

5) взаимосвязанные виды деятельности должны выполняться скоординированно. Координация улучшается при встраивании в структуру специальных инструментов взаимодействия.

Для изображения структурных взаимосвязей основных уровней и подразделений предприятия, их соподчиненности на практике используются схемы организационной структуры управления. Эти схемы являются только «скелетом» системы управления, поскольку не раскрывают состава и содержания функций, прав и обязанностей подразделений и должностных лиц.

В практике хозяйствования в зависимости от масштаба деятельности предприятия, его производственно-технических особенностей, стратегических и текущих задач может применяться несколько типов организационных структур:

- линейная;
- функциональная;
- линейно-функциональная;
- дивизиональная;
- проектная;
- матричная.

Линейная организационная структура управления — это структура, между элементами которой существует только

одноканальное взаимодействие. При такой структуре управления каждый подчиненный имеет лишь одного руководителя, который и выполняет все административные и специальные функции в соответствующем структурном подразделении. *Преимущества* организационной структуры управления линейного типа: четкость взаимоотношений, однозначность команд, оперативность подготовки и реализации управленческих решений, надежный контроль. *Недостаток* структуры заключается в том, что руководитель при этом должен быть высококвалифицированным универсалом, способным решать любые стратегические и текущие вопросы деятельности подчиненных ему подразделений (звеньев).

Основой **функциональной организационной структуры управления** является разделение функций управления между отдельными подразделениями аппарата управления (рис. 1.11).



Рис. 1.11. Пример функциональной структуры

Данное деление носит общий и неполный характер. Каждая функциональная структура должна учитывать специфику конкретной фирмы. Визитной карточкой функциональной структуры является централизация, профессионализм и экономичность [13].

Централизация. В ситуации когда каждый функциональный руководитель отвечает только за одну производственную функцию, ответственность за реализацию производственного процесса в целом лежит на генеральном директоре компании.

Профессионализм. Функциональная структура стимулирует профессиональную специализацию работников, обеспечивает рост их квалификации, облегчает задачу координации деятельности специалистов внутри департамента, а в результате обеспечивает быстроту и четкость решения профессиональных задач.

Экономичность. Данная структура, в отличие от других, включает дублирование функций в подразделениях организации.

Однако функциональная структура управления имеет и определенные *недостатки*: нарушение принципа единоначалия, противоречивость распоряжений, сложность координации деятельности управленческих служб, снижение оперативности работы органов управления. Работника в функциональной структуре управления отличает узкопрофессиональное видение проблем, ориентация на цели подразделения, в котором он работает, в ущерб общим целям организации. Данная структура не способствует развитию духа новаторства и предпринимательства.

Линейно-функциональная организационная структура управления опирается на распределение полномочий и ответственности по функциям управления и порядок принятия решений по вертикали. Она позволяет организовать управление по линейной схеме (директор → начальник цеха → мастер), а функциональные отделы аппарата управления предприятия лишь помогают линейным руководителям решать управленческие задачи. При этом линейные руководители не подчинены руководителям функциональных отделов аппарата управления. *Преимущество линейно-функциональной структуры* состоит в быстрой реализации управленческих решений вследствие иерархичности ее построения, что способствует повышению уровня специализации и эффективности работы функциональных служб, делает возможным необходимый маневр ресурсами. Такая структура является наиболее целесообразной при массовом производстве с устоявшимся ассортиментом продукции и незначительными эволюционными изменениями технологии ее изготовления. *Недостаток* состоит в том, что в условиях частых технологических изменений, обновления номенклатуры продукции использование этой структуры замедляет сроки подготовки и принятия управленческих решений, не обеспечивает надлежащей согласованности в работе функциональных отделов (подразделений).

Дивизиональная организационная структура управления базируется на углублении разделения управленческого труда. Применение данной структуры сопровождается процессом

децентрализации оперативных функций управления, осуществляемых производственными звеньями, и централизации общекорпоративных функций (принятия стратегических решений, маркетинговых исследований, финансовой деятельности и др.), которые концентрируются в высших звеньях администрации интегрированных предпринимательских структур.

Следовательно, при дивизиональной структуре каждое производственное подразделение корпорации (концерна) имеет собственную достаточно разветвленную структуру управления, обеспечивающую автономное его функционирование. Лишь стратегические функции управления централизованы на корпоративном уровне. При дивизиональной структуре управления группирование видов деятельности субъекта хозяйствования осуществляется с применением принципа разделения труда по целям. Это означает, что вокруг определенного производства формируется автономная организационная общность (рис. 1.12).

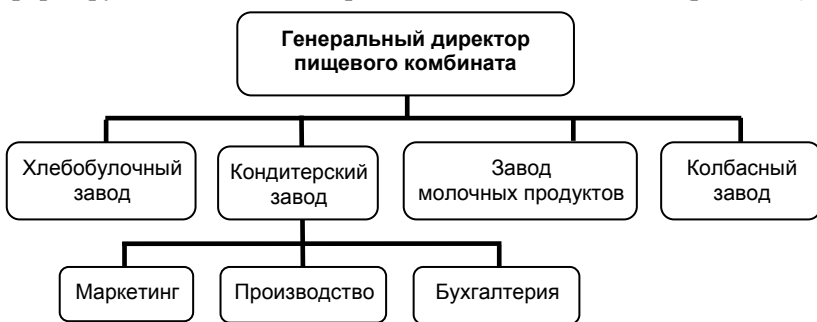


Рис. 1.12. Пример дивизиональной структуры

При дивизиональной организационной структуре управления возможны три способа группирования производственных подразделений [12]:

- 1) продуктовый (изготовление определенного продукта);
- 2) по группам потребителей (удовлетворение потребностей определенной группы потребителей);
- 3) по местонахождению (размещение в определенном географическом районе).

В сильно диверсифицированных корпорациях дивизионы формируются на основе видов бизнеса или стратегических бизнес-групп, объединяющих до нескольких десятков предприятий родственных сфер деятельности. При этом вводится должность президента стратегической бизнес-группы, функциями которого являются общее руководство, координация и контроль деятельности предприятий группы. Президент группы, с одной стороны, является проводником политики высшего руководства компании в своей группе, а с другой — отстаивает интересы своих предприятий в головном офисе компании, что обеспечивает необходимый баланс интересов в фирме. Кроме того, сокращается число руководителей подразделений, подотчетных непосредственно высшему руководству, что повышает управляемость компании.

Преимуществами дивизиональной организационной структуры управления являются гибкое реагирование на изменения во внешней среде, быстрое принятие управленческих решений и улучшение их качества. Корпоративное руководство освобождается от оперативных функций и имеет возможность сконцентрироваться на стратегии. *Недостатком* является необходимость увеличения при такой структуре численности аппарата управления и затрат на его содержание. Чем более независимы дивизиональные лидеры от высшего руководства, тем опаснее разрыв стратегического и оперативно-тактического руководства. Утратив контроль над ситуацией в дивизионах, руководство компании не в состоянии сформировать реалистическую общую стратегию, а лидеры дивизионов, лишенные ориентиров развития, могут принимать решения, противоречащие интересам фирмы в целом.

Вышерассмотренные организационные структуры управления (линейная, функциональная, линейно-функциональная, дивизиональная) являются жестко упорядоченными механическими структурами. С начала 60-х гг. XX в. получили распространение более гибкие типы организационных структур, способные модифицироваться в соответствии с изменениями внешней среды и потребностями самой организации. Такие структуры получили название адаптивных, или органических. Основными типами органических структур являются проектные и матричные структуры.

Проектная организационная структура возглавляется руководителем, который обеспечивает реализацию проекта в определенный срок и в пределах выделенной сметы. Для осуществления проекта формируется временная команда специалистов из представителей необходимых функциональных служб. Руководителю проекта полностью подчинены выделенные под его реализацию ресурсы и члены проектной команды. По завершении проекта команда распускается, специалисты начинают работать над новым проектом либо возвращаются в свои подразделения.

Преимуществом структуры является возможность полной концентрации членов проектной группы и ее руководителя на выполнении одного проекта. Данная структура — одна из наименее бюрократизированных: решения принимаются всей командой. Должность руководителя позволяет координировать действия с другими подразделениями. Проектная структура является идеальной организационной базой для решения проблем инновационного характера [13]. С проектной структурой связан ряд специфических *трудностей*:

- продолжительный период «настраивания» команды на работу, включающий периоды формирования команды, приспособления ее членов друг к другу и нормализации отношений внутри коллектива;
- проблема трудоустройства высвободившихся специалистов по завершении проекта или отказе от проекта;
- дублирование существующих в организации функциональных служб — самый существенный недостаток структуры.

В матричной организационной структуре управления наряду с линейными руководителями предприятия и функциональным аппаратом управления формируются еще и временные специализированные звенья (проектные группы), которые создаются из специалистов постоянных функциональных отделов и лишь временно подчиняются руководителю проекта. Принципиальным моментом в матричной структуре является подчинение сотрудников двум руководителям одного иерархического уровня. Кроме того, матричная структура — это постоянная структура, в отличие от проектной (рис. 1.13).

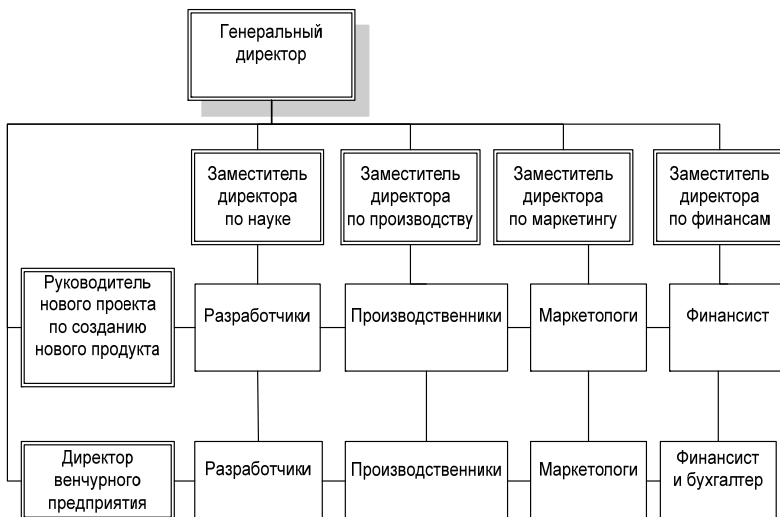


Рис. 1.13. Пример матричной организационной структуры

Руководитель проекта ставит задачи перед членами проектной команды, составляет графики выполнения задач, осуществляет контроль затрат, сроков, количественных и качественных показателей. Руководители функциональных служб определяют способ решения поставленных перед соответствующими специалистами задач и контролируют ход их решения.

Матричная структура обладает всеми *преимуществами* проектной структуры, притом что лишена некоторых ее недостатков. Она позволяет перераспределять функциональных специалистов между проектами, избегать дублирования функциональных служб, в ней отсутствует проблема трудоустройства специалистов после завершения проекта.

Однако матричные структуры управления имеют и определенные *недостатки*: увеличение численности управленческого персонала и количества информационных связей между работниками подразделений, возможные конфликтные ситуации между ними; психологические проблемы персонала, связанные с неопределенностью и переменчивостью структуры.

Новые формы производственных структур. Конец XX в. ознаменовался переходом от индустриального общества к информационному, для которого свойственны стремительные темпы изменения технологической, экономической и институциональной среды. На смену массовому производству стали приходиться гибкие производственные системы, одновременно обеспечивающие большие объемы производства и приспособленные к работе под заказ потребителя. Появились новые методы менеджмента, а следовательно, и новые производственные структуры [13]: модель «подтянутого производства», аутсорсинг, модель «горизонтальной корпорации», сетевая организационная структура.

*Модель «подтянутого производства»*¹ обеспечивает экономии затрат посредством автоматизации труда, компьютеризированного контроля над рабочими, «уплощения» производственной структуры, однако по существу производственную структуру она не меняет.

Широкое распространение получила вертикальная дезинтеграция корпораций, которая осуществлялась путем *аутсорсинга* — передачи субподрядов мелким и средним фирмам, отличающимся высокой производительностью и гибкостью. Использование сети субподрядных фирм придавало корпорации определенную гибкость, недостаточную, однако, для существования в условиях непрерывно меняющейся среды.

Создатели *модели «горизонтальной корпорации»*² полагали, что горизонтальные внутренние связи (координация деятельности подразделений) и горизонтальные внешние связи (кооперация) оказывают на эффективность работы больше влияния, чем традиционная вертикальная система управления, и в качестве организационных звеньев создали самоуправляемые комплексные рабочие группы.

Сетевая организационная структура является наиболее адекватной формой для новой экономики, как информационной, так и глобальной одновременно. Сетевая структура — это струк-

¹ Была апробирована в 1980-х гг. XX в.

² Была апробирована в 1990-х гг. XX в. рядом американских фирм.

тура, где установлены устойчивые отношения координации и взаимодействия между самоуправляемыми фирмами [13]. Сети возникают в случаях, когда для координации деятельности фирм уже недостаточно чисто рыночных механизмов, а интеграция в рамках единой корпорации не обеспечивает необходимой гибкости системы. Сетевая же структура отличается особой гибкостью. Сеть может изменять не только выпуск продукции в связи с непредвиденными обстоятельствами, но и собственную структуру — добавлять новые или исключать ненужные фирмы.

Примерами сетевых структур являются совместные предприятия, соглашения об участии в совместных исследовательских проектах и производственных программах, консорциумы, франчайзинговые соглашения и другое.

Сетевая кооперация позволяет фирмам разделять затраты и комбинировать ресурсы, также разделять риск неудачного технологического решения. Большинство сетей формируется вокруг крупного мультинационального предприятия либо на базе стратегических альянсов между такими предприятиями.

Одним из основных элементов бизнес-архитектуры, помимо организационной структуры, являются направленные на реализацию текущих задач и перспективных целей бизнес-процессы, описываемые функциональной моделью. Таким образом, **функциональная модель предприятия** представляет собой набор регламентов бизнес-процессов. Построение функциональной (процессной) модели предприятия — одна из самых важных задач для любого руководителя. С помощью функциональной модели можно полноценно строить техпроцесс, распределять ответственность за функции среди персонала, управлять контрольными точками, устанавливать категории и очередность процессов, собирать функции для формирования должностных инструкций и положений по подразделениям. В функциональной модели должно быть отражено как внутреннее взаимодействие бизнес-процессов, так и взаимодействие предприятия с внешней средой (рис. 1.14). При построении бизнес-процессов важна скорость построения процессов, сокращение рисков совершения ошибок, логический анализ процессов и многие другие факторы.

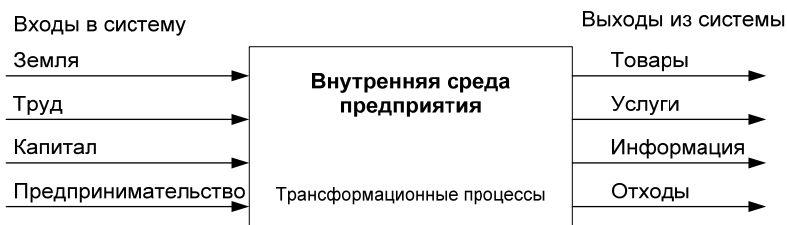


Рис. 1.14. Верхний уровень функциональной модели

При построении функциональных моделей бизнес-процессов может использоваться ряд терминов:

процесс (бизнес-процесс) — последовательность выполнения действий (функций), которые выполняются субъектами (ответственными за исполнение функций лицами) для достижения определенной задачи с определенным результатом. Каждый процесс входит в соответствующую категорию процессов и «имеет ответственную должность», т. е. выполнение процесса происходит под руководством и наблюдением ответственного лица. Процессы состоят из функций и взаимосвязаны друг с другом;

входящие процессы — процессы, ссылающиеся на данный процесс;

исходящие процессы — процессы, на которые ссылается данный процесс;

функции — составляющие процесса (шаг) — четырех видов:

1) **исполнительная функция**, требующая выполнения какого-либо действия персоналом любого уровня;

2) **функция анализа**, которая на основании анализа какого-либо действия может изменить дальнейшее выполнение процесса. Функция анализа имеет как минимум два выхода (решения). Эту функцию не может исполнять производственный персонал;

3) **контрольная функция**, которая контролирует или проверяет какое-либо действие. Контрольная функция не может выполняться производственным персоналом;

4) **руководящая функция**, разрешающая или утверждающая выполнение определенных действий, таких как издание прика-

зов, распоряжений и планов, подписание или утверждение документации, выбор стратегий, увольнение или прием персонала. Руководящая функция может завершать или открывать какой-то процесс или действие. Эту функцию может выполнять только руководящий персонал. Правила и нормы выполнения функции определяются соответствующим документом (инструкцией, методикой), частота выполнения функции фиксируется записью в таких документах, как акт, журнал, отчет и т. д.;

должности, подразделяющиеся на три типа:

1) *руководящая должность*, имеющая одного или нескольких подчиненных (например, директор или начальник отдела);

2) *дополнительная должность*, не участвующая в процессах производства или оказания услуг (например, главный бухгалтер);

3) *производственная должность*, принимающая участие в производстве продукции или оказании услуг и не имеющая подчиненных (например, монтажник или слесарь);

категория процесса — группа процессов, объединенных общей задачей (управление персоналом, маркетинг, управление производством). В отличие от процесса, за категорию процессов «отвечает» не должность, а, как правило, целое подразделение. Категориями процессов могут являться, например, следующие: маркетинг, логистика, производство, услуги; менеджменты проектов, персонала, качества, безопасности; административный, финансовый, экологический и менеджмент; управление рисками;

подкатегория процесса — дополнительный признак, различающий процессы, принадлежащие одной категории (например, в категории процесса «Финансовый менеджмент» выделяются такие подкатегории, как «Бюджетирование», «Бухгалтерский учёт», «Налоговый учёт» и т. д. Количество подкатегорий процессов в каждой категории не ограничивается;

документ — внутренний или внешний документ, используемый в деятельности компании (приказ или нормативный акт);

запись — документ, который ведется в деятельности предприятия, например журнал о регистрации корреспонденции;

дополнительная обязанность — внештатная единица (должность), которую из-за эпизодичности выполнения возложенных

на нее функций нельзя ввести в штатное расписание, например аудитор системы менеджмента качества или глава аттестационной комиссии. Периодически эти обязанности могут быть переложены с одного сотрудника на другого.

Построение функциональной модели начинается с обязательного описания в виде классификаторов следующих управленческих регистров:

- 1) бизнесов;
- 2) бизнес-процессов;
- 3) функций менеджмента;
- 4) организационной структуры компании.

После этого следует закрепить с помощью матричных проекций элементы первых трех классификаторов за четвертым, т. е. определить ответственность персонала за бизнесы, бизнес-процессы и функции менеджмента. Организационно-функциональная модель закрепляется Положением об организационной структуре, которое содержит описание указанных классификаторов и матричных проекций, дополненное граф-схемой организационной структуры предприятия.

Исходя из миссии формируются базовый рынок и базовый продукт предприятия, которые после соответствующей детализации (дифференцирования продукта и сегментации рынка) позволяют определить бизнесы в виде товарных групп, ориентированных на соответствующие рыночные сегменты.

Проекция бизнесов на этапы производственного цикла, включающего закупки сырья и комплектующих, производство продуктов и их распределение, обеспечивает формирование основных бизнес-процессов. Проекция компонентов производственного обеспечения на тот же производственный цикл позволит сформировать перечень обеспечивающих бизнес-процессов, носящих вспомогательный характер. Для этого сначала на предприятии разрабатываются и утверждаются два базовых классификатора: компоненты менеджмента и этапы управленческого цикла. После этого элементы указанных классификаторов проектируются друг на друга, порождая на пересечении строк и столбцов матрицы перечень основных функций менеджмента.

Каждое предприятие может разработать для себя персональные классификаторы. Принципиальное значение имеет не столько наполнение классификаторов, сколько единство их восприятия всеми менеджерами предприятия, закрепленное соответствующими внутрифирменными регламентами.

Закрепление функций за организационными звеньями производится согласно трафарету, приведенному на рис. 1.15. Зона ответственности помечается крестиком на пересечении соответствующей строки (звена) и столбца (функции).

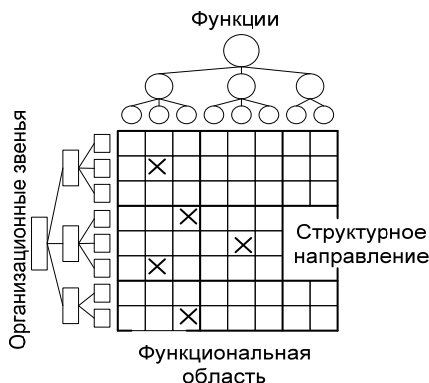


Рис. 1.15. Матрица распределения функций по организационным звеньям

Помимо корпоративной организационно-функциональной модели предприятия могут быть построены частные модели для отдельных структурных направлений (подразделений) или функциональных областей, например для отдела маркетинга или рекламного бюро, входящего в отдел маркетинга, и т. д. Значительно упрощает задачу использование для построения указанных моделей специальных программных систем, описание которых приведено в гл. 2–4 пособия.

Построение организационно-функциональной модели предприятия обеспечивает прозрачность и предсказуемость бизнеса, что повышает его инвестиционную привлекательность, открывает путь к партнерству и сотрудничеству.

Преимущества процессного подхода к управлению предприятием очевидны:

- четкое распределение ответственности за функции;
- прозрачность деятельности;
- возможность контролировать не только результаты, но и процесс их получения;
- возможность проведения осознанного анализа деятельности персонала;
- снижение зависимости от человеческого фактора;
- накопление базы знаний по выходу из кризисных ситуаций;
- широкие возможности совершенствования деятельности.

Однако не рекомендуется включать в функциональную модель программы единичные бизнес-процессы, которые не типичны для деятельности компании и выполняются всего один раз. Примером являются процессы, связанные с оформлением документов на открытие предприятия, или же процессы, выполняемые по индивидуальным требованиям заказчика при отсутствии возможности их упорядочивания и автоматизации. Но это вовсе не означает, что подобные процессы не стоит моделировать.

Построение единичных бизнес-процессов помогает внести ясность и согласованность между действиями сотрудников предприятия и представителями других организаций. Однако внесение их в процессную модель, наоборот, лишь усложняет понимание сотрудниками своих функциональных обязанностей. В таких бизнес-процессах сотрудники зачастую выполняют не характерные для своей должности функции, что также приводит к нарушению правил подчиненности, установленных в организационной структуре.

Не стоит хранить бизнес-процессы, выполнение которых наверняка больше никогда не повторится, и тем более пытаться управлять этими процессами. Это может привести к формированию чересчур раздутых должностных инструкций и получению неверных результатов при составлении отчетов.

1.2.4. Системная архитектура

В системную архитектуру (см. п. 1.2.1) включены следующие основные элементы [10]:

1) *архитектура информации:*

- базы данных и хранилища данных;
- системы управления БД или хранилищами данных;
- правила и средства санкционирования доступа к данным;

2) *архитектура приложений:*

- собственно прикладные системы, поддерживающие исполнение бизнес-процессов;
- интерфейсы взаимодействия прикладных систем между собой, с внешними системами, источниками или потребителями данных;
- средства и методы разработки и сопровождения приложений;

3) *технологическая архитектура:*

- *сетевая архитектура:*
 - локальные и территориальные вычислительные сети;
 - используемые в сетях коммуникационные протоколы, сервисы и системы адресации;
 - аварийные планы по обеспечению бесперебойной работы сетей в условиях чрезвычайных обстоятельств;
- *архитектура платформ:*
 - аппаратные средства вычислительной техники (серверы, рабочие станции, накопители и др.);
 - операционные и управляющие системы, утилиты и офисные программные системы;
 - аварийные планы по обеспечению бесперебойной работы аппаратуры (главным образом — серверов) и баз данных в условиях чрезвычайных обстоятельств.

Понятие архитектуры информации содержит описание создания возможностей быстрого принятия решений и распространения информации внутри организации и за ее пределами посредством использования информационных технологий. Можно

сказать, что архитектура информации является «зеркальным отражением» бизнес-архитектуры. Бизнес-архитектура отвечает на вопрос «кто и что будет делать с учетом общего видения, целей и стратегий?», а архитектура информации — на вопрос «какая информация должна быть предоставлена для осуществления этих процессов исполнителями?». Архитектура информации включает в себя модели, описывающие процессы обработки информации (information value chain), основные информационные объекты, связанные с бизнес-событиями, информационные потоки, принципы управления информацией. Архитектура должна описывать операционные данные, требуемые для выполнения процессов, аналитические данные и «контент», публикуемый в веб-приложениях.

Разработка архитектуры информации как части архитектуры предприятия состоит не в создании структур БД или моделей всех данных, использующихся предприятием. *Суть процесса разработки архитектуры информации* заключается в организации общего описания информации, требующейся для бизнеса, а также политики и правил работы с информацией. В связи с этим следует отметить, что в контексте архитектуры предприятия более правильно говорить об архитектуре и моделях информации, а не данных, хотя эти понятия и пересекаются. Модели архитектуры информации являются более абстрактными, они используют язык бизнеса и обеспечивают контекст, требуемый для моделирования данных. Модели данных предполагают четкие описания структуры объектов, атрибутов, отношений между сущностями.

Таким образом, понятие «архитектура информации» является расширением понятия «архитектура данных». В общем, ***архитектура информации*** рассматривается как процесс организации и представления значимой для бизнес-пользователей информации с использованием соответствующих средств каталогизации, навигации, пользовательского интерфейса. Этот аспект понятия архитектуры предприятия отражает место хранимой и обрабатываемой информации как стратегического корпоративного ресурса. Поэтому описание данной области будет дополнительно включать средства для оценки качества и востребованности данных, учета стоимости данных как нематериального актива и т. п.

В ходе разработки архитектуры информации решаются следующие задачи [1]:

- идентификация и инвентаризация существующих данных, включая определение их источников, процедур изменения и использования, ответственности, оценку качества;
- сокращение избыточности и фрагментарности данных с целью уменьшения их стоимости за счет снижения затрат на устройства хранения и их обслуживание, а также повышения качества данных за счет исключения неоднозначности и противоречивости различных экземпляров;
- исключение ненужных перемещений или копирования данных, особенно связанных с наличием большого количества унаследованных или устаревших приложений;
- формирование интегрированных представлений данных, таких как витрины и хранилища;
- обеспечение доступности данных в режиме, приближенном к режиму реального времени, за счет использования средств обмена сообщениями, интеграционных брокеров и шлюзов;
- интеграция метаданных, позволяющая обеспечить целостное представление данных из различных источников;
- сокращение числа используемых технологий и продуктов, ведущее к снижению расходов на обслуживание и появлению возможности получения дополнительных скидок от поставщиков применяемых продуктов;
- улучшение качества данных, прежде всего, за счет привлечения пользователей к управлению данными и их определению;
- улучшение защиты данных на основе использования последовательных и согласованных мер, обеспечивающих, с одной стороны, защиту от несанкционированного доступа, а с другой — доступность данных для их использования на практике.

Основные модели описания архитектуры информации.

Процессы разработки моделей информации и моделей данных предназначены для создания графических представлений информации, удовлетворяющей потребностям организации как в целом, так и в отдельных бизнес-процессах. Разработка таких

моделей является основой для реорганизации существующих бизнес-процессов и конструирования новых, которые будут использоваться для создания информационных систем. Анализ моделей информации проводится на различных уровнях абстракции: концептуальном, логическом и физическом.

На концептуальном уровне рассматриваются информационные потоки между функциональными подразделениями организации в обобщенном виде без описания практической реализации. Эти потоки не связаны с какой-либо автоматизированной системой и не описывают методы доступа или хранения информации.

На логическом уровне описываются требования к информации в форме и терминах, понятных бизнес-пользователям. Этот уровень анализа позволяет идентифицировать общие элементы данных, которые используются разными организационными подразделениями и бизнес-процессами, благодаря чему уменьшаются пересечения и сокращается возможность конфликтов между элементами данных. Назначение процесса моделирования на логическом уровне состоит в обеспечении средствами обнаружения, анализа, определения, стандартизации и нормализации отношений между бизнес-процессами и прикладными системами, идентификации потоков информации и соответствующих элементов данных, необходимых организации. Однако данный процесс не описывает способы хранения информации в базе данных.

На физическом уровне задается описание способов реализации логики бизнес-процесса соответствующей автоматизированной системы, приводится необходимый набор информационных объектов и их элементов данных. Физическая модель данных предназначена для представления логической модели в функции хранения в системе управления базами данных.

Архитектура приложений обеспечивает идентификацию прикладных систем, необходимых предприятию для выполнения бизнес-процессов, состоящих из этапов проектирования, разработки (или приобретения) и интеграции прикладных систем. Архитектура приложений предприятия состоит из двух частей: портфеля прикладных систем предприятия и области разработки прикладных систем.

Портфель прикладных систем представляет собой используемый на предприятии набор прикладных систем, обеспечивающих потребности бизнес-процессов предприятия. Он определяет область ответственности и приоритетность каждого приложения, а также способы достижения необходимой функциональности посредством либо разработки системы, либо покупки готовых приложений, аренды приложений или интеграции и использования возможностей уже имеющихся приложений. Портфель прикладных систем описывает приложения, предназначенные для выполнения функций организации и обмена информацией между клиентами, поставщиками и партнерами предприятия, каналы возможного взаимодействия пользователей с приложениями: web-браузеры, графический интерфейс «толстого» клиента, мобильные устройства и т. д.

Портфель прикладных систем формирует целостный взгляд на функциональные компоненты информационных систем, которые обеспечивают потребности бизнес-архитектуры и архитектуры информации и поддерживаются технологической архитектурой. Процессы управления портфелем прикладных систем тесно связаны с процессами управления ИТ-проектами и ИТ-активами в целом. Другими словами, **портфель прикладных систем** — это интегрированный набор информационных систем предприятия, который обеспечивает потребности бизнеса и включает в себя следующие компоненты [1]:

- *имеющийся портфель прикладных систем*. Это каталог имеющихся приложений и компонентов, который отражает их связи с поддерживаемыми ими бизнес-процессами, интерфейсы с другими системами, используемую и требуемую информацию, используемые инфраструктурные шаблоны. Чтобы быть реально полезным инструментом, он также должен помогать в идентификации тех элементов портфеля, которые можно использовать повторно и многократно в рамках предприятия, и стимулировать такое повторное использование;

- *планируемый портфель прикладных систем*, представляющий функциональность, которая требуется для обеспечения желаемого состояния бизнес-архитектуры и архитектуры

информации предприятия. Первым шагом в планировании портфеля прикладных систем является оценка текущего состояния портфеля и степени его соответствия потребностям организации со стратегической и технологической точек зрения. Соответствие бизнес-стратегиям оценивается на основе вклада прикладных систем в достижение бизнес-результатов, что определяется бизнес-архитектурой предприятия. Технологическое соответствие выявляется при сопоставлении прикладных систем, принципов и технологических стандартов, принятых в технологической архитектуре предприятия;

- *план миграции*. Процесс перехода от текущего к будущему портфелю прикладных систем в рамках ИТ-проектов. Проекты также могут объединяться в портфели проектов.

Область разработки прикладных систем описывает набор технологий, используемых для построения систем (разделение на функциональные составляющие, создание интерфейсов, настройку, а также используемые для этого шаблоны, руководства и т. д.). В данной области определяются процесс разработки и применяемые для него средства (программное обеспечение, средства проектирования), цикл разработки, контроль версий, настройка конфигураций. Основной задачей области разработки автоматизированных систем является уменьшение стоимости создания таких систем и повышение их качества за счет обеспечения единых подходов к разработке, что приводит к уменьшению общего количества различных технологических сценариев, связанных с проектированием архитектуры, операционной поддержкой, архитектурной интеграцией систем, обучением персонала. В этой области требуется участие системных архитекторов. Данная область выделяется, как правило, в организациях, в которых производится самостоятельная разработка приложений.

Внедрение на предприятии некоторой новой системы (например, биллинга) является частью управления портфелем прикладных систем предприятия, а технологии и принципы, используемые при проектировании системы, создании и сопровождении системы, относятся к области разработки [1].

Модели и инструменты управления портфелем приложений. Существуют различные способы оценки портфеля и различные классификации прикладных систем предприятия. Одним из возможных способов оценки портфеля прикладных систем является оценка по двум критериям — ценности с точки зрения бизнеса и техническому состоянию. Этот способ оценки получил название «*Матрица оценки состояния прикладных информационных систем (Health Grid)*» [14].

Оценка портфеля служит отправной точкой в идентификации проблемных областей и возможностей для лучшего удовлетворения потребностей бизнеса и принятия решения об инвестициях в новые системы или обновление существующих.

В результате такой оценки прикладные системы относят к одной из четырех возможных категорий (рис. 1.16) [1]:

1) *системы, находящиеся под угрозой вывода из эксплуатации (замены) или консолидации* вследствие низкой ценности для бизнеса и плохого технического состояния. Следует иметь в виду, что стоимость замены некоторых унаследованных и бэк-офисных систем может оказаться неоправданно высокой и будет иметь весьма ограниченную ценность с точки зрения бизнеса;

2) *системы, требующие переоценки или перепозиционирования* по причине низкой ценности для бизнеса, но в отличном техническом состоянии. Как правило, это прикладные системы, которые были недавно запущены в эксплуатацию в соответствии с рекомендациями, принятыми в рамках архитектуры предприятия. Однако объем и характер решаемых ими задач или ограниченность области применения в рамках каких-то узких организационных функций таковы, что их вклад в достижение ключевых бизнес-результатов незначителен. В этой ситуации рекомендуется провести идентификацию и анализ возможностей использования данных приложений или их компонентов в рамках остальных бизнес-процессов и организационных структур предприятия;

3) *требующие обновления системы*, представляющие высокую ценность для бизнеса, но в плохом техническом состоянии. Эти прикладные системы исправно обслуживают ключевые бизнес-функции, но создают существенные проблемы, когда речь идет

об эксплуатации и сопровождении этих систем, либо возникает необходимость использования информации из них, либо при необходимости интеграции данных систем с другими прикладными системами предприятия. Возможным выходом здесь является постепенный переход на использование более адаптивной архитектуры приложения (компонентного подхода, n-уровневой архитектуры, основанных на пересылке сообщений интерфейсов и т. д.);

4) *системы, требующие сопровождения и развития*, имеющие высокую ценность для бизнеса и в отличном техническом состоянии. Эти системы критически важны с точки зрения бизнеса и спроектированы в соответствии с современными представлениями об архитектуре прикладных систем.



Рис. 1.16. Оценка портфеля прикладных систем по критериям «Бизнес-ценность» и «Техническое состояние»

Для оценки портфеля прикладных систем может быть также использована *модель, предложенная компанией Gartner*. Анализ портфеля инвестиций может быть существенно упрощен, если взять за основу принцип ценности приложения для выполнения ключевых функций организации и цели, которые руководство преследует при внедрении соответствующих систем. Используя этот подход, высшие руководители организации могут

разделить портфель приложений на три класса в соответствии с относительным вкладом каждого приложения в выполнение ключевых функций и эффективность деятельности организации [1].

К *первому классу* относятся **базовые транзакционные** (вспомогательные или обслуживающие) приложения. Они играют важную роль с точки зрения обеспечения деятельности организации, но успех в выполнении критически важных задач и лучшие результаты по сравнению с другими организациями создают не они. Хорошими примерами являются приложение для расчета заработной платы или система управления персоналом. Операции, выполняемые этими системами, должны проводиться четко и вовремя, но например, сам факт своевременного получения сотрудником зарплаты еще не означает высокую эффективность работы организации в целом.

Важными требованиями к таким приложениям являются низкая стоимость, надежность, возможность выполнять большой объем операций при низкой стоимости в расчете на одну транзакцию. В действительности такие приложения в портфеле ИС предприятия составляют большинство.

Второй класс приложений — это **информационные приложения**, обеспечивающие преимущества бизнесу (предоставление информации для учета, управления, контроля, составления отчетов, анализа, совместной работы). Такими приложениями являются, например, системы предоставления отчета о продажах, аналитические системы. Использование данных приложений благоприятно сказывается на деятельности организации. Примерами преимуществ от использования ИС являются:

- ускорение цикла выполнения операций (например, принятия решения);
- быстрый вывод на рынок новых продуктов и услуг;
- уменьшение производственного цикла;
- более высокое качество;
- более широкий набор продуктов и услуг;
- более глубокая настройка на потребителя;
- меньшая стоимость выполнения операций.

Третий класс составляют **инновационные (стратегические) приложения**. В некоторых случаях использование информационных технологий может носить радикально новый, революционный характер с точки зрения влияния на функционирование организаций: способность кардинального изменения самой основы конкуренции и получение преимуществ. Примерами таких систем могут быть система электронной торговли через Интернет или система обслуживания кредитных карт банкоматами, которые в начале жизненного цикла этих технологий обеспечивали рост рынка компаниям, их внедрившим.

Анализ портфеля основан на том факте, что различные прикладные системы играют существенно различные роли в организации, и при выполнении этих ролей возникают различные управленческие проблемы [1]. Преимущества описанного подхода к управлению портфелем приложений при принятии решений высшими руководителями организации, не являющимися ИТ-профессионалами, заключаются в простоте, ясности и чувстве уверенности при использовании данного портфеля.

Следует отметить еще один класс инвестиций в информационные технологии, который необходимо рассматривать в совокупности с тремя классами прикладных систем, это — **технологическая архитектура (инфраструктура)**. Таким образом, строится «пирамида» из четырех классов активов, вокруг которых сосредоточены инвестиции в область информационных технологий. Управление портфелем данных активов составляет основу работы руководства департаментов ИТ предприятия (рис. 1.17).

Технологическая инфраструктура направлена на организацию гибкого и динамичного бизнеса, уменьшение стоимости использования ИТ, стандартизацию и интеграцию бизнеса. Для класса базовых транзакционных приложений характерно сокращение издержек и затрат, повышение производительности. Для информационных приложений, дающих преимущества бизнесу, основной эффект непосредственно связан с результативностью бизнеса: улучшением контроля, ускорением рабочих циклов, улучшением интеграции и получения информации.



Рис. 1.17. Четыре класса активов в портфеле ИТ

Для инновационных (стратегических) приложений основными задачами являются: улучшение роста продаж, организация конкурентных преимуществ, позиционирование на рынке, предоставление инновационных услуг, совершенствование взаимодействия с клиентами.

Основное назначение технологической архитектуры — обеспечение надежных ИТ-сервисов, предоставляемых в рамках всего предприятия в целом и координируемых централизованно, как правило, департаментами информационных технологий. Технологическая архитектура определяет набор принципов и стандартов, которые обеспечивают информационные руководства в отношении выбора и использования следующих технологий:

- аппаратных платформ;
- операционных систем;
- систем управления базами данных;
- средств разработки;
- языков программирования;
- сервисов электронной почты;
- систем безопасности;
- сетевой инфраструктуры и т. д.

Инфраструктурные сервисы в основном стандартизированы в рамках предприятия и используются сразу несколькими прикладными системами, расположенными над уровнем инфраструктурных сервисов и непосредственно обеспечивающими выполнение бизнес-процессов. При наличии необходимой инфраструктуры новые прикладные системы, которые потребуются предприятию для выполнения новых бизнес-процессов или реализации новых стратегий, могут быть созданы достаточно быстро и эффективно, что позволит повысить динамичность и гибкость предприятия. Одной из частных задач, решаемых в рамках данной архитектуры, является формирование «списка закупаемых технологий» [1].

Существует два принципиально отличных подхода к формированию технологической архитектуры. *Первый подход* заключается в перечислении используемых на предприятии стандартов и теоретически позволяет уменьшить зависимость предприятия от конкретных поставщиков. Однако уменьшение этой зависимости имеет ограниченный успех, поскольку замена одного продукта другим, поддерживающим один и тот же набор стандартов, как правило, оказывается невозможной или затруднительной. Поэтому с середины 1990-х годов большинство предприятий стали использовать *второй подход*, который связан, в конечном итоге, с перечислением конкретных продуктов и технологий.

Упорядоченный в рамках технологической архитектуры список продуктов и технологий дает реальные преимущества [1]:

- технический персонал должен поддерживать уровень знаний, связанных с меньшим количеством продуктов, что уменьшает затраты на содержание персонала и его обучение;
- прикладные системы легче интегрировать между собой, когда они имеют много общих технических аспектов. Хотя заметим, что список технологий и поставщиков не является все-таки самым важным инструментом интеграции данных и систем. Вопросы семантики и согласования форматов, например, гораздо более сложны и не решаются выбором одной технологии;
- предприятие может получить экономию на масштабах, приобретая технологии ограниченного количества поставщиков (например, скидки на лицензии);

- много усилий может быть сэкономлено на процессах закупок, поскольку, после того как технология однажды выбрана, последующие закупки не требуют затрат времени на длительное изучение альтернатив.

На рис. 1.18 приведен пример областей, категорий, стандартов и спецификаций технической справочной модели TRM FEAF (Federal Enterprise Architecture Framework) — технической справочной модели методики Федеральной архитектуры США [15].

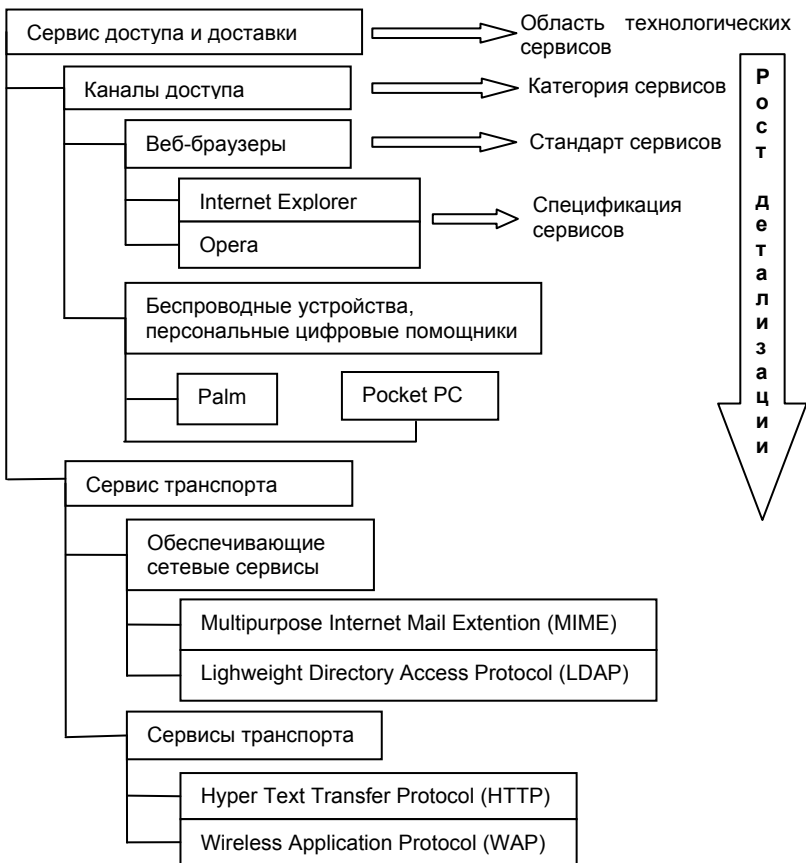


Рис. 1.18. Пример структуры технической справочной модели TRM FEAF

Большое значение имеет взаимосвязь между функциональными и операционными требованиями к системам и различными областями архитектуры, такими как прикладные системы и технологическая архитектура. Функциональные требования к прикладной системе описывают ценность, которую представляет система с точки зрения реализации функций организации (бизнес-ценность). Архитектура приложений, по сути, является архитектурой всех автоматизированных сервисов, которые обеспечивают и реализуют функциональные требования, включая интерфейсы к бизнес-приложениям и другим прикладным системам. Архитектура приложений описывает структуру приложений и способы реализации данной структурой функциональных требований организации [1].

Хорошая технологическая архитектура может обеспечивать безопасность, доступность, надежность и целый список других операционных требований, но если приложение спроектировано без использования преимуществ технологической архитектуры, оно все равно будет функционировать плохо, и его будет сложно внедрять и сопровождать. Аналогично, хорошо спроектированная структура прикладной системы, которая точно соответствует требованиям бизнес-процессов и собрана из многократно используемых компонентов с применением новейших технологий, может не соответствовать реальной конфигурации используемого аппаратного и системного программного обеспечения.

В настоящее время уделяется большое внимание понятию **«адаптивная технологическая инфраструктура»**. Имеется в виду технологическая инфраструктура, способная в определенных пределах, автоматически или полуавтоматически, «подстраиваться под требования» со стороны бизнес-приложений для обеспечения оптимальной работы. Основные идеи адаптивной инфраструктуры состоят в следующем:

- все ИТ-ресурсы являются общими и разделяемыми;
- выделение ресурсов конкретным приложениям производится автоматически в соответствии с требованиями бизнеса;
- качество обслуживания предсказуемо и стабильно, несмотря на непредсказуемый спрос на ресурсы.

Основными характеристиками адаптивной системы являются:

- *самоконфигурирование* — организация системы в соответствии с требованиями;
- *самозащита* — предотвращение сбоев в системе в результате нарушения работы компонентов системы и потери целостности данных;
- *самовосстановление* — диагностика неисправностей, локализация ошибок и устранение их последствий;
- *самооптимизация* — наиболее рациональное использование имеющихся ресурсов без вмешательства оператора.

Для реализации адаптивной технологической архитектуры предложили свои решения практически все ведущие производители, включая HP (концепция Adaptive Enterprise, архитектура Darwin), IBM (On Demand), Sun (N1), Microsoft (Dynamic Systems Initiative) и другие. Важной частью этих решений является комплексность, использующая как возможности аппаратных платформ, включая разделяемые процессорные разделы, виртуальные дисковые массивы, серверы, так и специализированное программное обеспечение для обработки существующих ресурсов.

Вопросы для самоконтроля

1. Дайте определение понятия «предприятие».
2. Сформулируйте необходимость изучения архитектуры предприятия.
3. Дайте определение понятия архитектуры предприятия.
4. Из каких элементов состоит архитектура предприятия при рассмотрении ее в статическом аспекте?
5. Из каких элементов состоит бизнес-архитектура предприятия?
6. Назовите базовые организационные структуры предприятия, их преимущества и недостатки.
7. Какие формы организационных структур возникли при переходе от индустриального общества к информационному?
8. Опишите процесс построения функциональной модели.

9. Из каких элементов состоит системная архитектура предприятия?
10. Из каких частей состоит архитектура приложений?
11. Приведите описание моделей архитектуры информации.
12. Что подразумевают под адаптивной технологической архитектурой?
13. Из каких элементов состоит архитектура предприятия при рассмотрении ее в динамическом аспекте?
14. Выделите слои в бизнес- и системной архитектуре предприятия.
15. Чем обусловлено значение архитектуры предприятия в современных условиях?

КЛАССИЧЕСКИЕ МЕТОДОЛОГИИ ПОСТРОЕНИЯ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ

2.1. Общие принципы построения архитектуры предприятия

В соответствии с основными положениями теории систем любой субъект, явление или процесс (включая предприятие) можно рассматривать как систему. Под **системой** понимают совокупность взаимосвязанных в одно целое элементов. *Элемент системы* — это часть целого, которая в процессе анализа не подлежит разделению на составляющие. Для предприятия как системы характерны следующие особенности:

1) *открытость*. Предприятие может существовать только при условии активного взаимодействия с внешней средой. Оно «выбирает» из промежуточной и общей внешней среды основные факторы производства, а затем, преобразовывая их в продукцию (товары, услуги, информацию) и отходы, направляет их во внешнюю среду. Условием жизнеспособности системы является полезный (выгодный) обмен между «входом» и «выходом»;

2) *искусственный принцип создания*. Предприятие является искусственной системой, созданной человеком ради собственных интересов, прежде всего совместного труда. Очевидной характеристикой любого предприятия является разделение труда, обуславливающее необходимость организации процессов управления.

Различают две формы разделения труда: горизонтальную и вертикальную. Первая — это разделение трудовых операций на отдельные задания. Результатом горизонтального разделения труда является формирование подразделений предприятия, выполняющих отдельные части общего трансформационного процесса. Вторая форма разделения труда связана с отделением работы по координированию действий от собственно действий. Это необходимо для достижения общей цели деятельности. Поэтому объективно возникает потребность в отделении управленческого труда от исполнительского. Таким образом, необходимость управления связана с процессами разделения труда на предприятии.

При разработке архитектуры предприятия приходится иметь дело с большим количеством измерений и связей между ними, которые необходимо учитывать. К настоящему моменту в мировой практике накоплен значительный опыт в области построения архитектуры предприятия. Существующие подходы и методы во многом базируются на использовании, обобщении и интеграции имеющихся результатов в области бизнес-моделирования, системного анализа и проектирования.

Построение архитектуры предприятия не является техническим процессом, связанным исключительно с информационными технологиями, хотя они и составляют достаточно весомую часть. На их основе разрабатываются программные системы, позволяющие создавать диаграммы и тексты, описывающие базовую информацию о деятельности организации и связывающие между собой различные факты. Применение ИТ помогает делать умозаключения, упрощающие и проясняющие процесс принятия сложных решений, повторяющийся в бизнесе каждый день. Специалисты и руководители, являющиеся пользователями методик построения архитектуры, составляют достаточно обширную аудиторию, в которой присутствуют:

- руководители предприятия;
- архитекторы;
- аналитики бизнес-процессов;
- системные аналитики;

- исполнители процессов;
- менеджеры (владельцы процессов).

Руководители предприятия осуществляют мониторинг внешней среды предприятия, на основе которого идентифицируют угрозы и новые возможности, высказывают предложения новых целей и стратегий.

Архитекторы предприятия на основании целей, стратегий и предложений от менеджеров производят идентификацию процессов, нуждающихся в изменениях, о чем сообщают бизнес-аналитикам и системным аналитикам. Роль архитектора характеризуется высоким статусом, отражающим степень важности организации архитектуры предприятия. Архитектор, как правило, является главным заместителем ИТ-директора и выступает постановщиком задач как для аналитиков бизнес-процессов, так и для системных аналитиков.

Бизнес-аналитики и системные аналитики реализуют изменения в процессах в тесном сотрудничестве с исполнителями процессов. Работу **исполнителей процессов** оценивают **менеджеры (владельцы процессов)** и руководители предприятия и вносят свои предложения архитекторам.

Основными этапами процесса построения архитектуры организации являются следующие [4]:

- 1) осознание необходимости построения архитектуры;
- 2) формирование рабочей группы;
- 3) выбор среды моделирования, средств моделирования и репозитория;
- 4) наполнение среды фактическим материалом (формирование архитектуры);
- 5) использование;
- 6) расширение и сопровождение.

На этапе **формирования архитектуры**, являющемся наиболее трудоемким, решаются задачи, относящиеся собственно к моделированию [4]:

- 1) определение бизнес-целей и требований;
- 2) моделирование бизнеса с позиции менеджера;

- 3) моделирование бизнес-процессов;
- 4) моделирование бизнес-функций;
- 5) моделирование оргструктуры, включая логические схемы принятия решений;
- 6) моделирование ресурсов;
- 7) преобразование бизнес-моделей в модели приложений и технологической архитектуры.

В основе современных подходов к построению моделей бизнес-слоя и системного слоя архитектуры предприятия лежат классические подходы, такие как методологии структурного и объектно-ориентированного анализа и проектирования, и интеграция разнообразных методов моделирования и анализа систем, проявляющаяся в форме создания интегрированных средств моделирования. В следующих параграфах этой главы приведено описание методов формирования архитектуры предприятия.

2.2. Методологии структурного анализа и проектирования

2.2.1. Структурный анализ

Структурным анализом принято называть метод исследования системы, которое начинается с ее общего обзора и затем детализируется, приобретая иерархическую структуру. Для методов данного класса характерны [4]:

- разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 6–7, при этом верхняя граница соответствует возможностям человеческого мозга по восприятию определенного количества взаимосвязанных объектов, а нижняя выбрана из соображений здравого смысла);
- ограниченный контекст, включающий лишь существенные на каждом уровне детали;
- использование строгих формальных правил записи;
- последовательное приближение к конечному результату.

Все методологии структурного анализа базируются на ряде общих принципов, регламентирующих организацию работ по моделированию. В качестве двух базовых принципов используются принцип «разделяй и властвуй» и принцип иерархического упорядочивания [4]. **Принцип «разделяй и властвуй»** используется при решении трудных проблем путем разбиения их на множество мелких независимых задач, легких для понимания и решения (так называемых «черных ящиков»). При этом пользователю не требуется знать, каким образом решается данная задача, необходимо лишь выявить входы и выходы «черного ящика», а также его назначение, т. е. функцию, которую он выполняет. **Принцип иерархического упорядочивания** облегчает проведение анализа системы при ее разбиении на части и, кроме того, «декларирует, что устройство этих частей также существенно для понимания».

Важным моментом в анализе сложных систем является широкое использование структурных методов графических нотаций: «одна картинка стоит тысячи слов» и аналитику такое представление системы понятно с первого взгляда. Структурные методы позволяют также дополнить графические изображения любой дополнительной информацией. Для достижения целей структурного анализа и проектирования традиционно используются средства, иллюстрирующие объекты графических нотаций:

- функции, которые система должна выполнять (более точно — функциональную структуру системы, отражающую последовательность выполняемых действий, передачу информации между элементами функциональной структуры);
- отношения между данными;
- динамическое поведение системы.

К структурным методологиям¹, применяемым для построения архитектуры предприятия, относят:

- методологию DFD (Data Flow Diagrams), предназначенную для построения диаграмм потоков данных;

¹ Методология — система базисных принципов, методов, методик, способов и средств их реализации в организации и построении научно-практической деятельности людей.

- методологию IDEF0 (Integration DEFinition), предназначенную для моделирования с использованием диаграмм функционального моделирования. Данная методология входит в группу методологий SADT (Structured Analysis and Design Technique);
- методологию IDEF3, предназначенную для документирования технологических процессов.

Данные методологии ориентированы на регламентацию моделирования бизнес- и системной архитектуры предприятия. В них отражается последовательность шагов, моделей и подходов, рациональное применение которых существенно улучшает результаты проектирования. Использование данных методологий помогает охватить и учесть все важные этапы и моменты разработки автоматизированных информационных систем, решить проблемы работы с большими объемами информации, сопровождающими процессы проектирования и координации действий коллектива, участвующего в проекте, а также оценить ход выполнения проекта. Применение описываемых методологий не гарантирует качества создаваемых моделей, их использование лишь оптимизирует и облегчает работу специалистов по разработке моделей.

Важнейшей характеристикой структурной методологии является порядок построения модели, в соответствии с которым методологии классифицируются на два вида — функционально-ориентированные и информационно-ориентированные. Традиционный функционально-ориентированный подход регламентирует первичность проектирования функциональных компонентов по отношению к проектированию структур данных: требования к данным раскрываются через функциональные требования. В информационно-ориентированном подходе наиболее важными являются вход и выход: структуры данных определяются первыми, а процедурные компоненты являются производными от данных. Предпочтительное использование функционально-ориентированных подходов связано с тем, что современная организация характеризуется переносом центра тяжести на слой бизнес-правил. Модель процесса является ценным средством для размышлений и совместной работы над перспективами развития организации и системной разработкой, поскольку руководство прекрасно ори-

ентируется в технологиях и бизнес-процессах организации и функциональные модели (в отличие от информационных) интуитивно понимаемы неспециалистами. Кроме того, информационная модель, как правило, представляет собой единственную диаграмму, которая может содержать несколько сотен объектов, тогда как функциональная иерархическая модель может включать десятки тысяч объектов.

Тем не менее информационная модель продолжает оставаться важной и соответствующим образом влиять на разрабатываемую функциональную модель. Подтверждением первичности функциональной модели является тот факт, что на Западе, где различные методики реорганизации применяются уже длительное время, большинство методологий являются функционально-ориентированными.

2.2.2. Методология на основе диаграмм потоков данных DFD

Диаграммы потоков данных представляют собой иерархию функциональных процессов, связанных потоками данных.

Процессы предназначены для продуцирования выходных потоков из входных в соответствии с действиями, задаваемыми именами процессов. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением, например «Сформировать личную карточку участника». Кроме того, каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Потоки данных — это механизмы, используемые для моделирования передачи информации (или, возможно, физических компонентов) из одной части системы в другую. Потоки на диаграммах обычно изображаются именованными стрелками, ориентация которых указывает направление движения информации.

Цель представления потоков данных в виде диаграмм — продемонстрировать, как каждый процесс преобразует входные

данные в выходные, а также выявить отношения между этими процессами. Классический стандарт DFD содержит набор символов или обозначений, с помощью которых описывается бизнес-процесс. Эти обозначения принято называть языком или методологией описания процессов.

Для построения диаграмм в методологии DFD традиционно используются нотации *Йордана Де Марко* и *Гейна-Сарсона*. Эти нотации незначительно отличаются друг от друга по графическому изображению символов.

В методе Гейна-Сарсона предложено классическую DFD-схему немного усложнить посредством введения дополнительного объекта, с помощью которого показываются места бизнес-процесса, где хранится информация либо материальные ресурсы. Примерами таких мест хранения являются:

- архив, в котором хранятся документы;
- база данных, в которой хранится информация;
- склад, на котором хранятся материальные ресурсы.

Данный объект получил название «*хранилище данных*». Хранилище данных позволяет на конкретных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет «срезы» потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое и быть существительным. В случае когда поток данных входит в хранилище или выходит из него и его структура соответствует структуре хранилища, он должен иметь то же самое имя, которое нет необходимости отражать на диаграмме.

В нотациях Гейна-Сарсона¹ и Йордона Де Марко² на DFD-схемах используются объекты, с помощью которых показывают внешних субъектов, взаимодействующих с бизнес-процессом. Данные объекты называют *внешними сущностями*. Имя сущ-

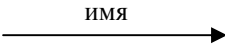
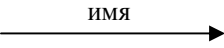


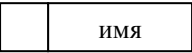
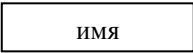


¹ Данная нотация реализована в пакете программ BPWin.

² Данная нотация реализована в пакете программ CaseAnalytic.

ности должно содержать существительное, например *склад товаров*. Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке. Основные символы DFD для различных нотаций изображены в табл. 2.1.

Таблица 2.1

Основные символы диаграммы потоков данных

Компонент	Нотация Гейна-Сарсона	Нотация Йордона Де Марко
Поток данных		
Процесс		
Хранилище данных		
Внешняя сущность		

Декомпозиция DFD осуществляется на основе процессов: каждый процесс может раскрываться с помощью DFD нижнего уровня [16].

Важную специфическую роль в модели играет специальный вид DFD — контекстная диаграмма, моделирующая систему наиболее общим образом. Контекстная диаграмма отражает интерфейс системы с внешним миром, а именно информационные потоки между системой и внешними сущностями, с которыми она должна быть связана. Она идентифицирует эти внешние сущности, а также, как правило, единственный процесс, отражающий главную цель или природу системы, насколько это возможно. И хотя контекстная диаграмма выглядит тривиальной, несомненная ее полезность заключается в том, что она устанавливает границы анализируемой системы. Каждый проект должен иметь ровно одну контекстную диаграмму, при этом нет необходимости в нумерации единственного ее процесса.

DFD первого уровня строится как декомпозиция процесса, который присутствует на контекстной диаграмме. Построенная диаграмма первого уровня также имеет множество процессов, которые, в свою очередь, могут быть декомпозированы в DFD нижнего уровня. Таким образом, строится иерархия DFD с контекстной диаграммой в корне дерева. Этот процесс декомпозиции продолжается до тех пор, пока процессы могут быть эффективно описаны с помощью коротких (до одной страницы) миниспецификаций обработки (спецификаций процессов).

При таком построении иерархии DFD каждый процесс более низкого уровня необходимо соотнести с процессом верхнего уровня. Обычно для этой цели используются структурированные номера процессов.

Так, например, если детализируется процесс номер 2 на диаграмме первого уровня и раскрывается данный процесс с помощью DFD, содержащей три процесса, то их номера будут иметь следующий вид: 2.1, 2.2 и 2.3. При необходимости можно перейти на следующий уровень, т. е. для процесса 2.2 получим 2.2.1, 2.2.2 и т. д.

На рис. 2.1. приведен пример DFD-схемы бизнес-процесса «Оформление и выдача трудовой книжки сотруднику при увольнении», разработанной в нотации Гейна-Сарсона [17]. Схема построена с использованием программного средства Microsoft Visio.

При сравнении рис. 2.1 и данных из табл. 2.1. можно увидеть незначительные отличия в отображении элементов нотаций. В современных программных средствах отображение классических нотаций может различаться.

Все сказанное по поводу диаграмм в нотации Гейна-Сарсона обнаруживается и в DFD-схеме на рис. 2.2, оформленной в нотации Йордона Де Марко. На этой схеме в качестве хранилища данных выступают сейф для хранения трудовых книжек и архив, в который помещается обходной лист, в качестве внешней сущности — увольняющийся сотрудник, получающий трудовую книжку как выход рассматриваемого бизнес-процесса.

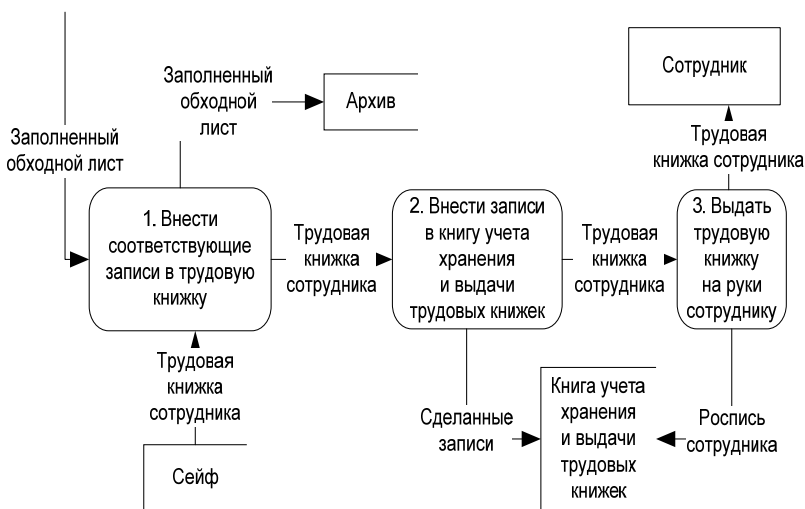


Рис. 2.1. DFD-схема бизнес-процесса «Оформление и выдача трудовой книжки сотруднику при увольнении» в нотации Гейна-Сарсона

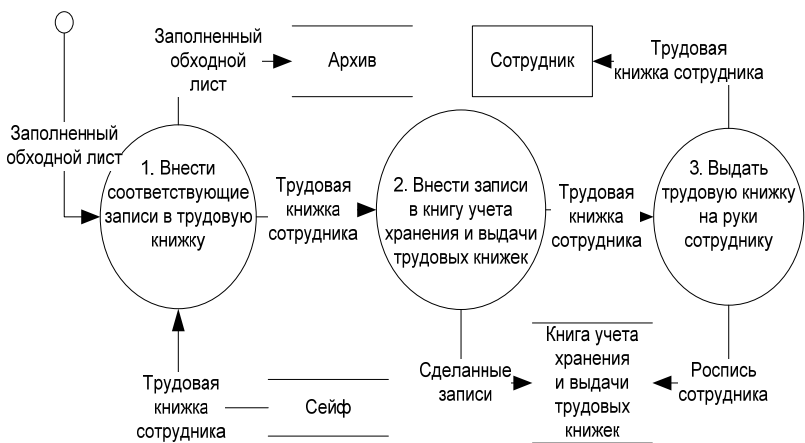


Рис. 2.2. DFD-схема бизнес-процесса «Оформление и выдача трудовой книжки сотруднику при увольнении» в нотации Йордона Де Марко

В первом приближении нотация Йордона Де Марко аналогична нотации Гейна-Сарсона, за исключением форм объектов: для описаний операций бизнес-процесса вместо закругленных прямоугольников стали использоваться круги, немного видоизменились и другие объекты — хранилище данных и внешние сущности (см. рис. 2.2) [17].

Практически любой класс систем успешно моделируется при помощи ориентированных DFD-методов. Они с самого начала создавались как средство проектирования информационных систем (тогда как SADT — как средство моделирования систем вообще) и имеют более богатый набор элементов, адекватно отражающих специфику таких систем (например, хранилища данных являются прообразами файлов или баз данных, внешние сущности отражают взаимодействие моделируемой системы с внешним миром).

2.2.3. Методология структурного анализа и проектирования IDEF0

Методология IDEF0 относится к группе методологий SADT. **SADT — методология структурного анализа и проектирования**, интегрирующая процесс моделирования, управление конфигурацией проекта, использование дополнительных языковых средств и руководство проектом со своим графическим языком. Процесс моделирования может быть разделен на несколько этапов: опрос экспертов, создание диаграмм и моделей, распространение документации, оценка адекватности моделей и принятие их для дальнейшего использования. Методология SADT возникла в конце 60-х гг. XX в. в ходе «революции», вызванной появлением структурного программирования. Когда большинство специалистов занимались созданием программного обеспечения, немногие старались разрешить более сложную задачу разработки крупномасштабных систем, включающих как людей и машины, так и программное обеспечение, аналогичных системам, применяемым в телефонной связи, промышленности, управлении и контроле за вооружением. В это время специалисты, традици-

онно занимавшиеся созданием крупномасштабных систем, стали осознавать необходимость большей упорядоченности, которая была достигнута путем формализации процесса создания системы и разбиения его на следующие фазы:

- 1) анализ — определение назначения системы;
- 2) проектирование — определение подсистем и их взаимодействие;
- 3) реализация — разработка подсистем по отдельности, объединение — соединение подсистем в единое целое;
- 4) тестирование — проверка работы системы;
- 5) установка — введение системы в действие;
- 6) эксплуатация — использование системы.

IDEF0 (Function Modeling) — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов, разработана Дугласом Россом. Отличительной особенностью IDEF0 является ее акцентирование на соподчиненности объектов. В данной методологии рассматриваются логические отношения между работами, а не их временная последовательность. Изначально модель IDEF0 была разработана как стандарт в 1981 г. в рамках программы автоматизации промышленных предприятий ICAM (Integrated Computer Aided Manufacturing), предложенной Департаментом военно-воздушных сил США. Собственно семейство стандартов IDEF унаследовало свое обозначение от названия этой программы (ICAM DEFinition). В процессе практической реализации участники программы ICAM столкнулись с необходимостью разработки новых методов анализа процессов взаимодействия в промышленных системах. При этом кроме усовершенствованного набора функций для описания бизнес-процессов одним из требований к новому стандарту было наличие эффективной методологии взаимодействия в рамках «аналитик – специалист». Другими словами, новый метод должен был обеспечить групповую работу над созданием модели с непосредственным участием всех аналитиков и специалистов, занятых в рамках проекта.

В результате поиска соответствующих решений появилась методология функционального моделирования IDEF0. С 1981 г.

стандарт IDEF0 претерпел несколько незначительных изменений, в основном ограничивающего характера, и последняя его редакция была выпущена в декабре 1993 г. Национальным институтом по стандартам и технологиям США (NIST).

Методология IDEF0 предписывает построение иерархической системы диаграмм — единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция: система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. Каждая диаграмма IDEF0 содержит блоки и дуги. *Блоки* изображают функции моделируемой системы. *Дуги* связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Функциональные блоки (работы) на диаграммах изображаются прямоугольниками, означающими поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Имя работы должно быть выражено отглагольным существительным, обозначающим действие. Согласно требованиям методологии IDEF0 необходимо, чтобы в диаграмме было не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, доступном для чтения, понимания и использования. Каждая сторона блока имеет особое, вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя — для управления, правая — для выходов, нижняя — для механизмов. Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, что и как выполняет функция.

Блоки в IDEF0 размещаются по степени важности, определяемой автором диаграммы. Выявление относительного порядка в соответствии со степенью важности блоков (работ), которая ха-

характеризуется влиянием одного блока на другой, составляет сущность процесса доминирования. Доминирующим блоком диаграммы может быть либо первый из требуемой последовательности функций, либо планирующая или контролирующая функция, влияющая на все другие. Доминирующий блок размещается в верхнем левом углу диаграммы, а наименее важный — в правом углу.

Расположение блоков отражает авторское определение доминирования. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные. Чтобы подчеркнуть это, аналитик может перенумеровать блоки в соответствии с порядком их доминирования. Порядок доминирования обозначается цифрой, размещенной в правом нижнем углу каждого прямоугольника: цифра 1 указывает на преобладающее влияние какого-либо блока на другие, цифра 2 и последующие характеризуют работы (блоки) по степени уменьшения влияния.

Взаимодействие работ с внешним миром и между собой описывается в виде стрелок, изображаемых одинарными линиями со стрелками на концах. Стрелки представляют собой некую информацию и именуются существительными.

В IDEF0 различают пять типов стрелок:

1) *вход* — объекты, используемые и преобразуемые работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Стрелка входа рисуется как входящая в левую грань работы;

2) *управление* — информация, управляющая действиями работы. Обычно управляющие стрелки несут информацию, которая указывает сущность выполняемой работы. Каждая работа должна иметь хотя бы одну стрелку управления, которая изображается как входящая в верхнюю грань работы;

3) *выход* — объекты, в которые преобразуются входы. Каждая работа должна иметь хотя бы одну стрелку выхода, которая рисуется как исходящая из правой грани работы;

4) *механизм* — ресурсы, выполняющие работу. Стрелка механизма рисуется как входящая в нижнюю грань работы. По усмотрению аналитика стрелки механизма могут не изображаться на модели;

5) *вызов* — специальная стрелка, указывающая на другую модель работы. Стрелка вызова рисуется как исходящая из нижней части работы и отражает тот факт, что некоторая работа выполняется за пределами моделируемой системы.

Для описания отношений между блоками в методологии IDEF0 требуется пять типов взаимодействий:

- 1) управление;
- 2) вход;
- 3) обратная связь по управлению;
- 4) обратная связь по входу;
- 5) выход–механизм.

Связи по управлению и входу являются простейшими, поскольку они отражают прямые воздействия, которые интуитивно понятны и очень просты. Отношение управления возникает в случае, когда выход одного блока непосредственно влияет на блок с меньшим доминированием. Обратная связь по управлению и обратная связь по входу являются более сложными, поскольку представляют собой итерацию или рекурсию. А именно выходы из одной работы влияют на будущее выполнение других работ, что впоследствии повлияет на исходную работу. Обратная связь по управлению возникает, когда выход некоторого блока влияет на блок с большим доминированием.

Связи «выход–механизм» встречаются нечасто. Они отражают ситуацию, при которой выход одной функции становится средством достижения цели для другой. Связи «выход–механизм» характерны при распределении источников ресурсов (например, требуемых инструментов, обученного персонала, физического пространства, оборудования, материалов).

В IDEF0 дуга редко изображает один объект. Обычно она символизирует набор объектов. Так как дуги представляют наборы объектов, они могут иметь множество начальных точек (источников) и конечных точек (назначений). Поэтому дуги могут разветвляться и соединяться различными способами. Вся дуга или ее часть может выходить из одного или нескольких блоков и заканчиваться в одном или нескольких блоках. Разветвление дуг, изображаемое в виде расходящихся линий, озна-

чает, что все содержимое дуг или его часть может появиться в каждом ответвлении. Дуга всегда помечается до разветвления, чтобы дать название всему набору. Кроме того, каждая ветвь дуги может быть помечена или не помечена в соответствии со следующими правилами:

- непомеченные ветви содержат все объекты, указанные в метке дуги перед разветвлением;
- ветви, помеченные после точки разветвления, содержат все объекты или их часть, указанные в метке дуги перед разветвлением.

Слияние дуг в IDEF0, изображаемое как сходящиеся вместе линии, указывает, что содержимое каждой ветви идет на формирование метки для дуги, являющейся результатом слияния исходных дуг. После слияния результирующая дуга всегда помечается для указания нового набора объектов, возникшего после объединения. Кроме того, каждая ветвь перед слиянием может помечаться или не помечаться в соответствии со следующими правилами:

- непомеченные ветви содержат все объекты, указанные в общей метке дуги после слияния;
- помеченные перед слиянием ветви содержат все или некоторые объекты из перечисленных в общей метке после слияния.

Диаграмма IDEF0 верхнего уровня приведена на рис. 2.3.

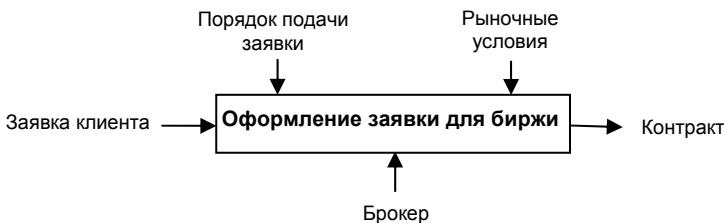


Рис. 2.3. Пример диаграммы IDEF0 верхнего уровня

Далее перечислим *рекомендации по желательным значениям факторов диаграммы*:

- 1) необходимо стремиться к тому, чтобы количество блоков на диаграммах нижних уровней было ниже количества блоков

на родительских диаграммах, т. е. чтобы с увеличением уровня декомпозиции убывал коэффициент.

Таким образом, убывание этого коэффициента говорит о том, что по мере декомпозиции модели функции должны упрощаться, следовательно, количество блоков должно убывать;

2) диаграммы должны быть сбалансированы. Например, у работы число входящих стрелок и стрелок управления не должно быть существенно большим, чем число выходящих стрелок.

Следует отметить, что данная рекомендация может не выполняться в моделях, описывающих производственные процессы: при описании процедуры сборки в блок может входить множество стрелок, описывающих компоненты изделия, а выходить одна стрелка — готовое изделие;

3) помимо анализа графических элементов диаграммы необходимо рассматривать наименования блоков. Для оценки имен составляется словарь элементарных (тривиальных) функций моделируемой системы. Фактически в данный словарь должны попасть функции нижнего уровня декомпозиции диаграмм.

Например, для модели БД элементарными могут являться функции «*найти запись*», «*добавить запись в БД*», в то время как функция «*регистрация пользователя*» требует дальнейшего описания;

4) после формирования словаря и составления пакета диаграмм системы необходимо рассмотреть нижний уровень модели. Если на нем обнаружатся совпадения названий блоков диаграмм и слов из словаря, то это свидетельствует о достижении достаточного уровня декомпозиции.

2.2.4. Методология моделирования и стандарт документирования процессов IDEF3

IDEF3 является стандартом документирования технологических процессов, происходящих на предприятии, и предоставляет инструментарий для наглядного исследования и моделирования их сценариев. Сценарием (Scenario) можно назвать описание

последовательности изменений свойств объекта в рамках рассматриваемого процесса (например, описание последовательности этапов обработки детали в цеху и изменение ее свойств после прохождения каждого этапа). Исполнение каждого сценария сопровождается соответствующим документооборотом, который состоит из двух основных потоков: документов, определяющих структуру и последовательность процесса (технологических указаний, описаний стандартов и т. д.), и документов, отображающих ход его выполнения (результатов тестов и экспертиз, отчетов о браке и т. д.). Для эффективного управления любым процессом необходимо иметь детальное представление о его сценарии и структуре сопутствующего документооборота. Средства документирования и моделирования IDEF3 позволяют [18]:

- документировать имеющиеся данные о технологии процесса, выявленные, например, в процессе опроса компетентных сотрудников, отвечающих за организацию исследуемого процесса;
- определять и анализировать точки влияния потоков сопутствующего документооборота на сценарий технологических процессов;
- определять ситуации, в которых требуется принятие решения, влияющего на жизненный цикл процесса, например изменение конструктивных, технологических или эксплуатационных свойств конечного продукта;
- содействовать принятию оптимальных решений при реорганизации технологических процессов;
- разрабатывать имитационные модели технологических процессов по принципу «Как будет, если...».

Существуют **два типа диаграмм в стандарте IDEF3**, представляющие описание одного и того же сценария технологического процесса в разных ракурсах:

- 1) диаграммы PFDD (Process Flow Description Diagrams) — диаграммы описания последовательности процессов;
- 2) диаграммы OSTN (Object State Transition Network) — диаграммы описания состояния объекта и его трансформаций в процессе.

Предположим, требуется описать процесс окраски детали в производственном цехе на предприятии. С помощью диаграмм PFDD документируется последовательность и описание стадий обработки детали в рамках исследуемого технологического процесса. Диаграммы OSTN используются для иллюстрации трансформаций детали на каждой стадии обработки.

Далее показывается, как графические средства IDEF3 позволяют документировать вышеуказанный производственный процесс окраски детали. Этот процесс состоит непосредственно из самой окраски, производимой на специальном оборудовании, и этапа контроля ее качества, который определяет, нужно ли деталь окрасить заново (в случае несоответствия стандартам и выявления брака) или отправить ее в дальнейшую обработку.

На рис. 2.4 [18] изображена диаграмма PFDD, являющаяся графическим отображением сценария обработки детали.

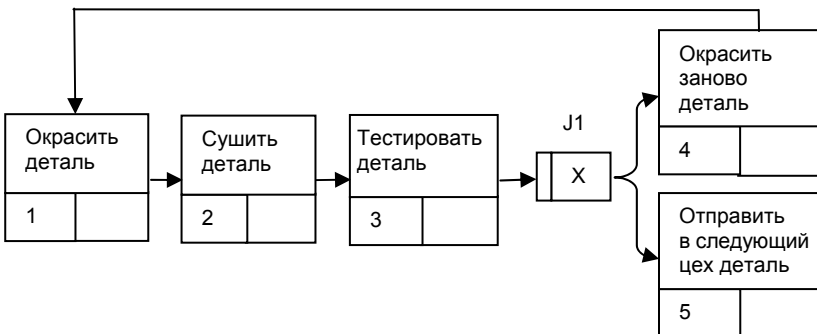


Рис. 2.4. Пример PFDD-диаграммы

Прямоугольники на диаграмме PFDD называются функциональными элементами или элементами поведения (Unit of Behavior, UOB) и обозначают событие, стадию процесса или принятия решения. Каждый UOB имеет свое имя, отображаемое в глагольном наклонении, и уникальный номер. Стрелки или линии являются отображением перемещения детали между UOB-блоками в ходе процесса.

Линии бывают следующих видов:

- старшая (Precedence) — сплошная линия, связывающая UOB. Рисуются слева направо или сверху вниз;
- отношения (Relational Link) — пунктирная линия, используемая для изображения связей между UOB;
- потоки объектов (Object Flow) — стрелка с двумя накопниками используется для описания факта использования объекта (детали) в двух или более единицах работы, например, когда объект порождается в одной работе и используется в другой.

Объект, обозначенный J1, называется перекрестком (Junction). Перекрестки используются для отображения логики взаимодействия стрелок (потоков) при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы. Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и разветвления. При внесении перекрестка в диаграмму необходимо указать тип перекрестка. Классификация возможных типов перекрестков приведена в табл. 2.2 [18]. Все перекрестки в PFDD-диаграмме нумеруются, каждый номер имеет префикс «J».

Сценарий, отображаемый на диаграмме, можно описать в следующем виде [18]: «Деталь, подготовленная к окраске, поступает в окрасочный цех. В процессе окраски наносится один слой эмали при высокой температуре. После этого производится сушка детали, после которой начинается этап проверки качества нанесенного слоя. Если тест подтверждает недостаточное качество нанесенного слоя (недостаточную толщину, неоднородность и т. д.), то деталь заново пропускается через цех окраски. Если деталь успешно проходит контроль качества, то она отправляется в следующий цех для дальнейшей обработки».

Каждый функциональный блок UOB может иметь последовательность декомпозиций и, следовательно, может быть детализирован с любой необходимой точностью. Под декомпозицией понимается представление каждого UOB с помощью отдельной IDEF3-диаграммы. Например, можно декомпозировать UOB

«Окрасить деталь», представив его отдельным процессом и построив для него свою PFDD-диаграмму. При этом данная диаграмма будет называться дочерней по отношению к диаграмме, изображенной на рис. 2.4, а та, соответственно, — родительской.

Таблица 2.2

Классификация возможных типов перекрестков

Обозначение	Наименование	Слияния стрелок (Fan-in Junction)	Разветвления стрелок (Fanout Junction)
	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Synchronous AND	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Synchronous OR	Один или несколько предшествующих процессов завершаются одновременно	Один или несколько следующих процессов запускаются одновременно
	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

УОВ дочерних диаграмм имеют сквозную нумерацию, т. е. если родительский УОВ имеет номер 1, то блоки УОВ на его декомпозиции будут соответственно иметь номера 1.1, 1.2 и т. д. Применение принципа декомпозиции в IDEF3 позволяет структурированно описывать процессы с любым требуемым уровнем детализации.

Если PFDD-диаграммы позволяют рассматривать технологический процесс «с точки зрения наблюдателя», то другой класс диаграмм IDEF3 — OSTN-диаграммы — предоставляет

возможность анализировать тот же самый процесс «с точки зрения объекта». На рис. 2.5 представлено отображение процесса окраски с точки зрения OSTN-диаграммы [18].

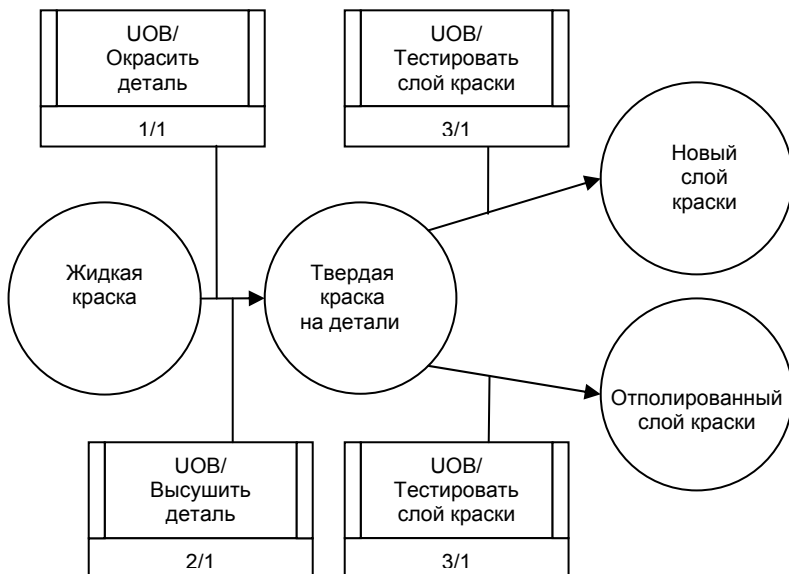


Рис. 2.5. Пример OSTN-диаграммы

Состояния объекта (детали) и изменение состояния являются ключевыми понятиями OSTN-диаграммы. Состояния объекта отображаются окружностями, а их изменения — направленными линиями. Каждая линия имеет ссылку на соответствующий функциональный блок UOB, в результате действий которого произошло отображаемое ей изменение состояния объекта.

Следует особо отметить, что вышеперечисленные три разновидности структурного анализа и проектирования (DFD, IDEF0, IDEF3) примерно одинаковы с точки зрения возможностей изобразительных средств функционального моделирования. При этом одним из *основных критериев выбора того или иного метода* является степень владения им консультантом или аналитиком,

грамотность выражения своих мыслей на языке моделирования. В противном случае в моделях, построенных с использованием любого метода, будет невозможно разобраться.

2.2.5. Методология моделирования отношений между данными IDEF1X

Следующим этапом после выполнения этапа структурного анализа системы является процесс проектирования. Здесь встает вопрос качественного моделирования отношений между данными. Наиболее распространенным средством моделирования отношений между данными (информационного моделирования) является диаграмма «сущность-связь» ERD (Entity-Relationship Diagram), известная в двух нотациях — Чена и Баркера. Она предназначена для обеспечения стандартного способа определения данных и отношений между ними, с ее помощью документируются информационные аспекты системы, включая идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их связей с другими объектами (отношений). ERD традиционно используется в структурном анализе и проектировании, однако, по существу, представляет собой подмножество объектной модели предметной области.

Одна из разновидностей модели «сущность-связь» используется в методе IDEF1/IDEF1X, входящем в семейство стандартов IDEF. **Методология IDEF1X** предназначена для разработки реляционных баз данных с использованием условного синтаксиса, специально разработанного для построения концептуальной схемы проектируемой системы¹. Методологию IDEF1X целесообразно использовать и для построения архитектуры информации на логическом уровне, после того как проведено исследование всех информационных ресурсов и принято решение о применении реляционной базы данных (средства моделирования IDEF1X специально разработаны для построения схем реляционных БД).

¹ Построение архитектуры информации на концептуальном уровне как части системной архитектуры рассмотрено в первой главе.

Существует несколько причин, по которым IDEF1X не следует применять при построении нереляционных систем [19]:

1) IDEF1X требует от проектировщика определения ключевых атрибутов, чтобы отличить одну сущность от другой, в то время как объектно-ориентированные системы не требуют задания ключевых атрибутов для идентифицирования объектов;

2) если несколько атрибутов однозначно идентифицируют сущность, проектировщик должен определить один из этих атрибутов в качестве первичного ключа, а все остальные определяются как вторичные. Модель IDEF1X, построенная и переданная для окончательной реализации, некорректна в плане применения методов объектно-ориентированной реализации и предназначена для построения реляционной системы.

На основании архитектуры информации логического уровня в средствах проектирования схем IDEF1X может быть осуществлен переход на физический уровень с указанием непосредственно системы управления базами данных.

Сущности в IDEF1X и их атрибуты¹. Сущность в IDEF1X описывает совокупность или набор экземпляров похожих по свойствам, но однозначно отличаемых друг от друга по одному или нескольким признакам. Каждый экземпляр является реализацией сущности. Таким образом, сущность в IDEF1X описывает конкретный набор экземпляров реального мира. Примером сущности IDEF1X может быть сущность СОТРУДНИК, которая представляет собой всех сотрудников предприятия, а один из них, например Иванов Петр Сергеевич, является конкретной реализацией этой сущности. В примере диаграммы IDEF1X (рис. 2.6) каждый экземпляр сущности СОТРУДНИК содержит следующую информацию: идентификатор сотрудника (ID сотрудника), имя сотрудника, адрес сотрудника и т. п. В IDEF1X-модели эти свойства называются атрибутами сущности. Каждый атрибут содержит только часть информации о сущности.

¹ Описание элементов методологии IDEF1X приведено с использованием материалов статьи Г. Верникова [19].

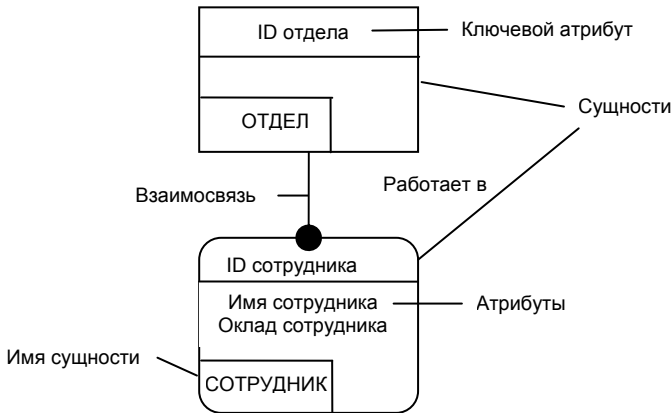


Рис. 2.6. Диаграмма связи между сотрудником и отделом

Связи в IDEF1X представляют собой ссылки, соединения и ассоциации между сущностями. Связи — это глаголы, которые показывают, как соотносятся сущности между собой. Ниже приведен ряд примеров связей между сущностями:

- ОТДЕЛ <состоит из> нескольких СОТРУДНИКОВ;
- САМОЛЕТ <перевозит> нескольких ПАССАЖИРОВ;
- СОТРУДНИК <пишет> разные ОТЧЕТЫ.

Во всех перечисленных примерах взаимосвязи между сущностями соответствуют схеме «один ко многим». Это означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности. Причем первая сущность называется родительской, а вторая — дочерней. В приведенных примерах глаголы заключены в угловые скобки. Связи отображаются в виде линии между двумя сущностями с точкой на одном конце и глагольной фразой, отображаемой над линией. На рис. 2.6 приводится диаграмма связи между СОТРУДНИКОМ и ОТДЕЛОМ.

Отношения «многие ко многим» обычно используются на начальной стадии разработки диаграммы, например в диаграмме зависимости сущностей, и отображаются в IDEF1X в виде сплошной линии с точками на обоих концах. Так как отношения «мно-

гие ко многим» могут скрыть другие бизнес-правила или ограничения, они должны быть полностью исследованы на одном из этапов моделирования. Иногда отношение «многие ко многим» на ранних стадиях моделирования идентифицируется неправильно, на самом деле представляя два или несколько случаев отношений «один ко многим» между связанными сущностями. В случае необходимости хранения дополнительных сведений о связи «многие ко многим» (например, даты или комментария) такая связь должна быть заменена дополнительной сущностью, содержащей эти сведения. При моделировании необходимо быть уверенным в том, что все отношения «многие ко многим» будут подробно обсуждены на более поздних стадиях моделирования для обеспечения правильного моделирования отношений.

Идентификация сущностей. Представление о ключах.

Сущность описывается в диаграмме IDEF1X графическим объектом в виде прямоугольника. На рис. 2.7 приведен пример IDEF1X-диаграммы.

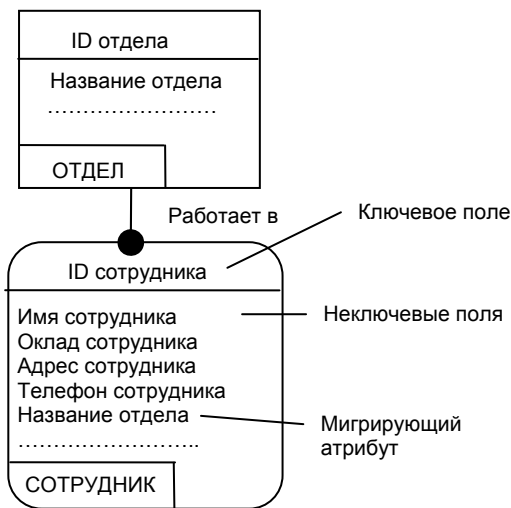


Рис. 2.7. Диаграмма связи между сотрудником и отделом с мигрирующим атрибутом

Каждый прямоугольник, отображающий собой сущность, разделяется горизонтальной линией на часть, в которой расположены ключевые поля, и часть, где расположены неключевые поля. Верхняя часть называется ключевой областью, а нижняя часть — областью данных. Ключевая область сущности СОТРУДНИК содержит поле «Уникальный идентификатор сотрудника», в области данных находятся поля «Имя сотрудника», «Адрес сотрудника», «Телефон сотрудника» и т. д.

Ключевая область содержит первичный ключ для сущности. Первичный ключ — это набор атрибутов, выбранных для идентификации уникальных экземпляров сущности. Атрибуты первичного ключа располагаются над линией в ключевой области. Как следует из названия, неключевой атрибут — это атрибут, который не был выбран ключевым. Неключевые атрибуты располагаются под чертой, в области данных.

При создании сущности в модели IDEF1X одним из главных является вопрос о способе идентификации уникальной записи. Для этого требуется уникальная идентификация каждой записи в сущности, позволяющая правильно создать логическую модель данных. Напомним, что сущности в IDEF1X всегда имеют ключевую область, и поэтому в каждой сущности должны быть определены ключевые атрибуты.

Выбор первичного ключа для сущности является очень важным шагом и требует большого внимания. В качестве первичных ключей могут быть использованы несколько атрибутов или групп атрибутов. Атрибуты, которые могут быть выбраны первичными ключами, называются кандидатами в ключевые атрибуты (потенциальные атрибуты). Кандидаты в ключи должны уникально идентифицировать каждую запись сущности. В соответствии с этим ни одна из частей ключа не может быть NULL, т.е. незаполненной или отсутствующей. Например, для того чтобы корректно использовать сущность СОТРУДНИК в IDEF1X-модели данных (а позже — в базе данных), необходимо иметь возможность уникально идентифицировать записи. Правила выбора первичного ключа из списка предполагаемых ключей достаточно строги, однако могут быть применены ко всем типам баз данных

и информации. В соответствии с правилами атрибуты и группы атрибутов должны отвечать следующим требованиям:

- уникальным образом идентифицировать экземпляр сущности;
- не использовать NULL-значений;
- не изменяться со временем, поскольку экземпляр идентифицируется при помощи ключа, при изменении которого меняется и экземпляр;
- быть наиболее короткими для использования индексирования и получения данных. Если вам нужно использовать ключ, являющийся комбинацией ключей из других сущностей, убедитесь в том, что каждая из частей ключа соответствует правилам.

Выбор первичных ключей показан на примере выбора таковых для ранее рассмотренной сущности СОТРУДНИК. При выборе необходимо учитывать следующее:

- атрибут «ID сотрудника» является потенциальным ключом, поскольку он уникален для всех экземпляров сущности СОТРУДНИК;
- атрибут «Имя сотрудника» не рекомендуется использовать в качестве потенциального ключа, так как среди служащих на предприятии может быть, к примеру, два Ивана Петрова;
- атрибут «Номер страхового полиса сотрудника» является уникальным, но СОТРУДНИК может не иметь такового;
- комбинация атрибутов «Имя сотрудника» и «Дата рождения сотрудника» может быть использована в качестве искомого потенциального ключа.

После проведенного анализа можно назвать два потенциальных ключа — атрибут «Номер сотрудника» и комбинацию, включающую поля «Имя сотрудника» и «Дата рождения сотрудника». Атрибут «Номер сотрудника» имеет самые короткие и уникальные значения, поэтому лучше других подходит для первичного ключа.

При выборе первичного ключа для сущности разработчиками модели часто используется дополнительный (суррогатный) ключ — произвольный номер, уникальным образом определяю-

щий запись в сущности. Атрибут «Номер сотрудника» является примером суррогатного ключа и лучше всего подходит на роль первичного ключа, потому что является коротким и быстрее всего идентифицирует экземпляры в объекте. Кроме того, суррогатные ключи могут автоматически генерироваться системой так, чтобы нумерация была сплошной, без пропусков. Потенциальные ключи, не выбранные в качестве первичных, могут быть использованы как вторичные (альтернативные) ключи. С их помощью часто отображают различные индексы доступа к данным в конечной реализации реляционной базы.

Если сущности в IDEF1X-диаграмме связаны, связь передает ключ (или набор ключевых атрибутов) дочерней сущности. Эти атрибуты называются внешними ключами. Внешние ключи определяются как атрибуты первичных ключей родительского объекта, переданные дочернему объекту через их связь. Передаваемые атрибуты называются мигрирующими. На рис. 2.7 таким атрибутом является атрибут «Название отдела».

Классификация сущностей в IDEF1X. При разработке модели зачастую приходится сталкиваться с сущностями, уникальность которых зависит от значений атрибута внешнего ключа. Для уникального определения каждой сущности внешний ключ должен быть частью первичного ключа дочернего объекта. Дочерняя сущность, уникальность которой зависит от атрибута внешнего ключа, называется *зависимой сущностью*. В примере на рис. 2.6 сущность СОТРУДНИК является зависимой сущностью, потому что ее идентификация зависит от сущности ОТДЕЛ. В обозначениях IDEF1X зависимые сущности представлены в виде закругленных прямоугольников. Зависимые сущности далее классифицируются на сущности, которые не могут существовать без родительской сущности, и сущности, которые не могут быть идентифицированы без использования ключа родителя (сущности, зависящие от идентификации). Сущность СОТРУДНИК принадлежит ко второму типу зависимых сущностей, так как сотрудники могут существовать и без отдела.

Имеются ситуации, в которых сущность зависит от существования другой сущности. Рассмотрим две сущности: ЗАПРОС,

которая используется для отслеживания запросов покупателей, и ПОЗИЦИЯ ЗАПРОСА, отслеживающая отдельные элементы в сущности ЗАПРОС. Связь между этими двумя сущностями может быть выражена в следующем виде: ЗАПРОС <содержит> один или несколько ПОЗИЦИЙ ЗАПРОСА. В этом случае ПОЗИЦИЯ ЗАПРОСА зависит от существования ЗАКАЗА.

Сущности, не зависящие при идентификации от других объектов в модели, называются *независимыми сущностями*. В описанном примере сущность ОТДЕЛ можно считать независимой. В IDEF1X независимые сущности представлены в виде прямоугольников. В IDEF1X концепция зависимых и независимых сущностей усиливается типом взаимосвязей между двумя сущностями. Если необходимо, чтобы внешний ключ передавался в дочернюю сущность (и в результате создавал зависимую сущность), то *создается идентифицирующая связь* между родительской и дочерней сущностями. Идентифицирующие взаимосвязи обозначаются сплошной линией между сущностями.

Неидентифицирующие связи, являющиеся уникальными для IDEF1X, также связывают родительскую сущность с дочерней. Они используются для отображения другого типа передачи атрибутов внешних ключей — передачи в область данных дочерней сущности (под линией). Эти связи отображаются пунктирной линией между объектами. Так как переданные ключи в неидентифицирующей связи не являются составной частью первичного ключа дочерней сущности, то этот вид связи не проявляется ни в одной идентифицирующей зависимости. В этом случае и ОТДЕЛ, и СОТРУДНИК рассматриваются как независимые сущности (см. рис. 2.7). Тем не менее взаимосвязь может отражать зависимость существования, если бизнес-правило определяет ситуацию, при которой внешний ключ не может принимать значение NULL. Если внешний ключ должен существовать, то это означает, что запись в дочерней сущности может существовать только при наличии ассоциированной с ним родительской записи.

2.3. Методология объектно-ориентированного анализа и проектирования

2.3.1. Объектная модель

Важное место в разработках архитектурных моделей занимают объектно-ориентированные методологии, основанные на объектной декомпозиции предметной области, представляемой в виде совокупности объектов, взаимодействующих между собой посредством передачи сообщений. Объектно-ориентированная методология основывается на так называемой **объектной модели**, главными принципами построения которой являются: *абстрагирование, инкапсуляция, модульность, иерархичность, типизация, параллелизм, сохраняемость*. Каждый из принципов сам по себе не нов, но в объектной модели впервые принципы применены в совокупности. Объектно-ориентированный анализ и проектирование принципиально отличаются от традиционных подходов структурного проектирования: здесь по-другому трактуется процесс декомпозиции, а архитектура получающегося программного продукта в значительной степени выходит за рамки представлений, традиционных для структурного программирования.

Объектно-ориентированное проектирование, согласно определению Г. Буча, являющегося основоположником в данной области, — «это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы» [3].

Методы структурного проектирования помогают упростить процесс разработки сложных систем за счет использования алгоритмов как готовых строительных блоков. Аналогично, методы объектно-ориентированного проектирования созданы, чтобы помочь разработчикам применять мощные выразительные средства объектного и объектно-ориентированного программирования, использующего в качестве блоков классы и объекты.

В качестве объектов предметной области могут рассматриваться конкретные предметы, а также абстрактные или реальные сущности (например, клиент, заказ, организация и т. п.). Каждый объект характеризуется набором атрибутов, значения которых определяют его 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.

Основные принципы разработки объектной модели сформулированы и подробно рассмотрены в работе Гради Буча [3].

Абстрагирование является одним из основных принципов, используемых для решения сложных задач. Многие авторы дают различные по формулировке, но общие по смыслу определения абстракции. Проанализировав эти определения, Г. Буч вывел следующее заключение: «**Абстракция** выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов, и таким образом четко определяет его концептуальные границы с точки зрения наблюдателя». По мнению Е. Сейдвица и М. Старка, «существует целый спектр абстракций, начиная с объектов, которые почти точно соответствуют реалиям предметной области, и кончая объектами, не имеющими право на существование» [20]. Эти абстракции можно проанжировать от наиболее полезных к наименее полезным:

- *абстракция сущности* — объект представляет собой полезную модель некой сущности в предметной области;
- *абстракция поведения* — объект состоит из обобщенного множества операций;
- *абстракция виртуальной машины* — объект группирует операции, которые либо вместе используются более высоким уровнем управления, либо сами используют некоторый набор операций более низкого уровня;
- *произвольная абстракция* — объект включает в себя набор операций, не имеющих друг с другом ничего общего.

«Инкапсуляция — это процесс отделения друг от друга элементов объекта, определяющих его устройство и поведение; инкапсуляция служит для того, чтобы изолировать контрактные обязательства абстракции от их реализации». Абстракция и инкапсуляция дополняют друг друга: абстрагирование направлено на наблюдаемое поведение объекта, а инкапсуляция занимается внутренним устройством. Чаще всего инкапсуляция выполняется посредством скрытия информации (маскировки всех внутренних деталей, не влияющих на внешнее поведение). Обычно скрываются и внутренняя структура объекта, и реализация его методов.

Суть процесса инкапсуляции можно раскрыть на примере гидропонного тепличного хозяйства. В [20] отмечается, что «... одной из ключевых абстракций данной предметной области является нагреватель, поддерживающий заданную температуру в помещении. Нагреватель является абстракцией низкого уровня, что позволяет ограничиться всего тремя действиями с данным объектом: включение, выключение и запрос состояния. Нагреватель не должен отвечать за поддержание температуры, это будет поведением более высокого уровня, совместно реализуемым нагревателем, датчиком температуры и еще одним объектом. Мы говорим о поведении *более высокого уровня*, потому что оно основывается на простом поведении нагревателя и датчика, добавляя к ним кое-что еще, а именно *гистерезис* (или запаздывание), благодаря которому можно обойтись без частых включений и выключений нагревателя в состояниях, близких к граничным. Приняв такое решение о разделении ответственности, мы делаем каждую абстракцию более цельной».

«Модульность — это свойство системы, которая была разложена на внутренне связанные, но слабо связанные между собой модули». Разделение программы на модули до некоторой степени позволяет уменьшить ее сложность и создать хорошо определенные и документированные интерфейсы. Таким образом, принципы абстрагирования, инкапсуляции и модульности являются взаимодополняющими. Объект логически определяет границы определенной абстракции, а инкапсуляция и модульность делают их физически незывлемыми [3].

«**Иерархия** — это упорядочение абстракций, расположение их по уровням». Абстракция — вещь полезная, но всегда, кроме самых простых ситуаций, число абстракций в системе намного превышает наши умственные возможности. Инкапсуляция позволяет в какой-то степени устранить это препятствие, убрав из поля зрения внутреннее содержание абстракций. Модульность также упрощает задачу, объединяя логически связанные абстракции в группы. Но этого оказывается недостаточно. Значительное упрощение в понимании сложных задач достигается за счет образования из абстракций иерархической структуры.

Иерархия может выражаться несколькими типами:

- одиночное наследование (например, дом есть недвижимость, станок есть оборудование);
- множественное наследование (например, на предприятии есть две должности — юрист и бухгалтер, сотрудник И. Иванов может занимать обе должности);
- агрегация (например, сотрудники отдела маркетинга находятся в состоянии агрегации с отделом маркетинга).

Понятие *типа* взято из теории абстрактных типов данных. «**Типизация** — это способ защититься от использования объектов одного класса вместо другого, или, по крайней мере, управлять таким использованием». Суть типизации состоит в выражении используемых абстракций таким образом, чтобы язык программирования, применяемый в реализации, поддерживал соблюдение принятых проектных решений.

«**Параллелизм** — это свойство, отличающее активные объекты от пассивных». Существуют задачи, которые должны обрабатываться одновременно, т. е. параллельно. При параллелизме главное внимание уделяется абстрагированию и синхронизации процессов. Каждый объект, полученный из абстракции реального мира, может представлять собой отдельный поток управления (активный поток). В качестве примера параллелизма можно привести реализацию какого-либо производственного процесса, при котором нужно производить периодический контроль протекания процесса (измерение температуры в доменных печах, слежение за курсом самолета и т. д.).

«**Сохраняемость** — способность объекта существовать во времени, переживая породивший его процесс, и (или) в пространстве, перемещаясь из своего первоначального адресного пространства». Данный принцип более тесно связан с объектно-ориентированным программированием, нежели с проектированием.

В заключение *приведем основные преимущества объектной модели:*

1) возможность в полной мере использовать выразительные средства объектных и объектно-ориентированных языков программирования;

2) существенное повышение уровня унификации разработки и пригодность модели для повторного использования не только программ, но и проектов, что в конце концов ведет к созданию среды разработки;

3) построение систем на основе стабильных промежуточных описаний, значительно упрощающих процесс внесения изменений. Это дает системе возможность развиваться постепенно и не приводит к полной ее переработке даже в случае существенных изменений исходных требований;

4) уменьшение риска при разработке сложных систем, прежде всего потому, что процесс интеграции растягивается на все время разработки, а не превращается в единовременное событие. Объектный подход состоит из ряда хорошо продуманных этапов проектирования, что также уменьшает степень риска и повышает уверенность в правильности принимаемых решений;

5) ориентированность модели на человеческое восприятие мира. По мнению Д. Робсона, «многие люди, не имеющие понятия о том, как работает компьютер, находят вполне естественным объектно-ориентированный подход к системам» [21].

2.3.2. Язык моделирования UML

Большинство современных методов объектно-ориентированного подхода основано на использовании унифицированного языка моделирования UML (Unified Modeling Language), являюще-

гося фактически преемником наиболее распространенных объектно-ориентированных методов Буча, Рамбо и Якобсона. UML представляет собой язык для определения, представления, проектирования и документирования программных систем, организационно-экономических систем, технических систем и других систем различной природы. UML находится в процессе стандартизации, осуществляемой OMG (Object Management Group) — организацией по стандартизации в области объектно-ориентированных методов и технологий. В настоящее время язык UML принят в качестве стандартного языка моделирования и имеет широкую поддержку в индустрии программного обеспечения. Стандарт UML версии 1.1, принятый OMG в 1997 г., содержит структурные модели и модели поведения в виде набора диаграмм.

В целом, интегрированная модель сложной системы в нотации UML может быть представлена в виде совокупности диаграмм, представленных на рис. 2.8–2.16 [22].



Рис. 2.8. Интегрированная модель сложной системы в нотации UML

Структурные (structural) модели содержат:

диаграммы классов (class diagrams) — для моделирования статической структуры классов системы и их связей (рис. 2.9);

диаграммы компонентов (component diagrams) — для моделирования иерархии компонентов (подсистем) системы (рис. 2.10);

диаграммы развертывания/размещения (deployment diagrams) — для моделирования физической архитектуры системы (рис. 2.11).



Рис. 2.9. Примеры графического изображения конкретных классов

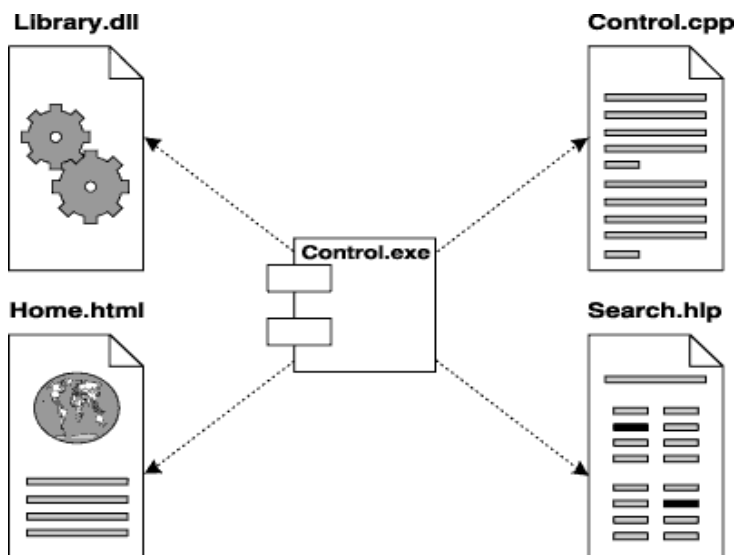


Рис. 2.10. Графическое изображение отношения зависимости между компонентами

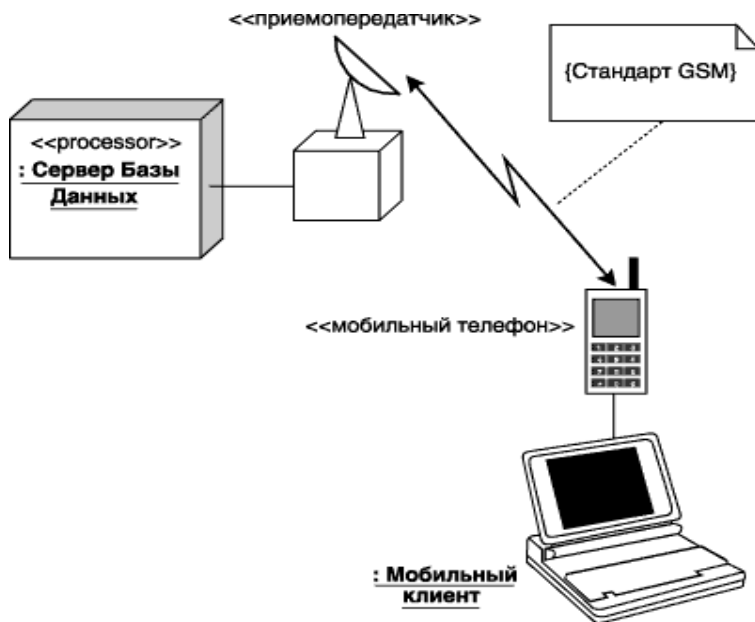


Рис. 2.11. Диаграмма развертывания для системы мобильного доступа к корпоративной базе данных

Модели поведения (behavioral) включают:

диаграммы вариантов использования (use case diagrams) — для моделирования функциональных требований к системе (в виде сценариев взаимодействия пользователей с системой) (рис. 2.12);

диаграммы взаимодействия (interaction diagrams):

– диаграммы последовательности (sequence diagrams) — для моделирования последовательности действий (рис. 2.13);

– диаграммы кооперации (collaboration diagrams) — для моделирования процесса обмена сообщениями между объектами (рис. 2.14);

диаграммы состояний (statechart diagrams) — для моделирования поведения объектов системы при переходе из одного состояния в другое (рис. 2.15);

диаграммы деятельности (activity diagrams) — для моделирования поведения системы в рамках различных вариантов использования, или потоков управления (рис. 2.16).



Рис. 2.12. Диаграмма вариантов использования для системы продажи товаров по каталогу в общих обозначениях языка UML

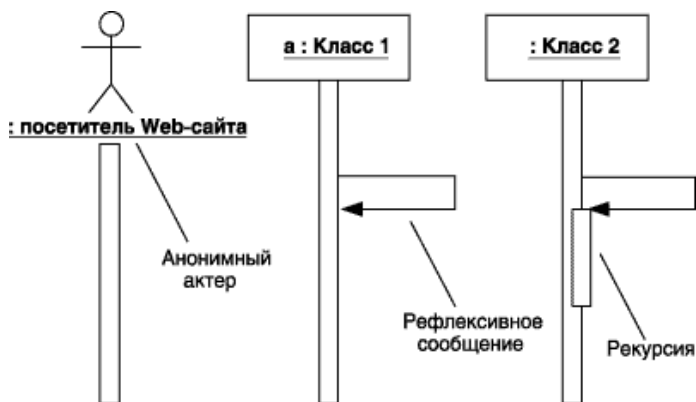


Рис. 2.13. Графическое изображение актера, рефлексивного сообщения и рекурсии на диаграмме последовательности



Рис. 2.14. Фрагмент диаграммы кооперации для выбора адреса клиента при отправке электронного письма

UML обладает механизмами расширения для адаптации языка моделирования к конкретным нуждам. Наличие механизмов расширения принципиально отличает UML от таких средств моделирования, как IDEF0, IDEF1, IDEF3, DFD. Перечисленные языки моделирования можно определить как сильно типизированные, поскольку они не допускают произвольной интерпретации семантики элементов моделей. UML, допуская такую интерпретацию, является слабо типизированным языком.

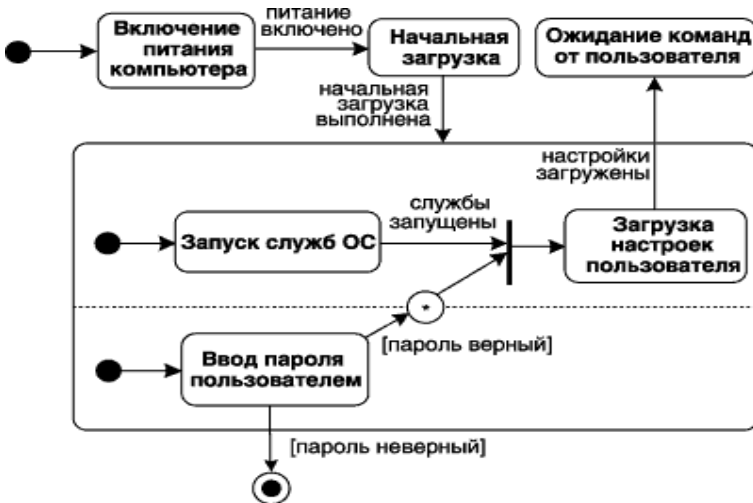


Рис. 2.15. Диаграмма состояний для примера включения компьютера

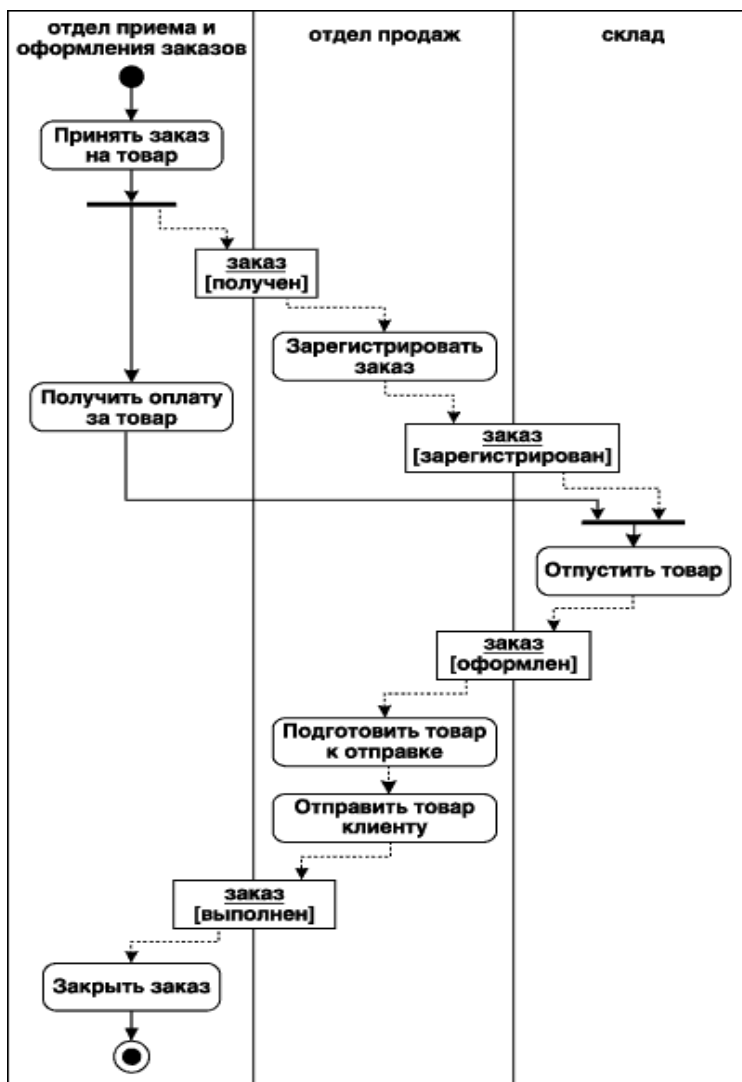


Рис. 2.16. Фрагмент диаграммы деятельности торговой компании с объектом-заказом

Основой взаимосвязи между структурным и объектно-ориентированным подходами является общность ряда категорий и понятий этих подходов (процесс и вариант использования, сущность и класс и др.). Эта взаимосвязь может проявляться в различных формах. Так, одним из возможных вариантов является использование структурного анализа как основы для объектно-ориентированного проектирования. При этом структурный анализ следует прекращать при переходе от бизнес-слоя к системному слою архитектуры. Другой формой проявления взаимосвязи можно считать интеграцию объектной и реляционной технологий.

Реляционные СУБД являются на сегодняшний день основным средством реализации крупномасштабных баз данных и хранилищ данных. Причины этого достаточно очевидны [4]:

- реляционная технология используется достаточно долго, освоена огромным количеством пользователей и разработчиков, стала промышленным стандартом, в нее вложены значительные средства и создано множество корпоративных БД в самых различных отраслях;
- реляционная модель проста и имеет строгое математическое основание;
- существует большое разнообразие промышленных средств проектирования, реализации и эксплуатации реляционных БД.

Вследствие этого реляционные БД в основном используются для хранения и поиска объектов в так называемых объектно-реляционных системах.

2.3.3. Паттерны

При моделировании бизнес-процессов и проектировании прикладных систем в различных проектах достаточно часто встречаются аналогичные ситуации. Во многих системах по проектированию и разработке программных систем реализованы такие понятия, как «шаблоны» («образцы»). Методология объектно-ориентированного анализа и проектирования не является исключением, в ней механизмы «шаблоны» из абстрактного понятия стали неотъемлемыми атрибутами. В языке моделирования UML, в

частности, они получили название «паттерны». Паттерны различаются по степени детализации и уровню абстракции. В [22] предлагается классификация паттернов по категориям их применения.

Архитектурные паттерны (Architectural patterns) — множество предварительно определенных подсистем со спецификацией их ответственности, правил и базовых принципов установления отношений между ними. Архитектурные паттерны предназначены для спецификации фундаментальных схем структуризации программных систем. Наиболее известными паттернами этой категории являются паттерны GRASP (General Responsibility Assignment Software Pattern), относящиеся к уровню системы и подсистем, но не к уровню классов. Как правило, они формулируются в обобщенной форме, используют обычную терминологию и не зависят от области приложения.

Паттерны проектирования (Design patterns) — специальные схемы для уточнения структуры подсистем или компонентов программной системы и отношений между ними. Они описывают общую структуру взаимодействия элементов программной системы, реализующих исходную проблему проектирования в конкретном контексте. Наиболее известные паттерны этой категории — паттерны GoF (Gang of Four), названные в честь Э. Гаммы, Р. Хелма, Р. Джонсона и Дж. Влассидеса, которые систематизировали их и представили общее описание. Паттерны GoF включают в себя 23 паттерна. Эти паттерны не зависят от языка реализации, но их реализация зависит от области приложения.

Паттерны анализа (Analysis patterns) — специальные схемы для представления общей организации процесса моделирования. Паттерны анализа относятся к одной или нескольким предметным областям и описываются в терминах предметной области. Наиболее известными паттернами этой группы являются паттерны бизнес-моделирования ARIS (Architecture of Integrated Information Systems), которые характеризуют абстрактный уровень представления бизнес-процессов. В дальнейшем паттерны анализа конкретизируются в типовых моделях с целью выполнения аналитических оценок или имитационного моделирования бизнес-процессов.

Паттерны тестирования (Test patterns) — специальные схемы для представления общей организации процесса тестирования программных систем. К этой категории паттернов относятся такие паттерны, как тестирование черного ящика, белого ящика, отдельных классов, системы. Некоторые из них реализованы в инструментальных средствах, наиболее известными из которых являются IBM Test Studio. В связи с этим паттерны тестирования иногда называют стратегиями или схемами тестирования.

Паттерны реализации (Implementation patterns) — совокупность компонентов и других элементов реализации, используемых в структуре модели при написании программного кода. Эта категория паттернов делится на следующие подкатегории:

- паттерны организации программного кода;
- паттерны оптимизации программного кода;
- паттерны устойчивости кода;
- паттерны разработки графического интерфейса пользователя и др.

Паттерны проектирования в нотации языка UML. В сфере разработки программных систем наибольшее применение получили паттерны проектирования *GoF*, некоторые из них реализованы в популярных средах программирования. При этом паттерны проектирования могут быть представлены в наглядной форме с помощью рассмотренных обозначений языка UML. Паттерн проектирования в контексте языка UML представляет собой параметризованную кооперацию вместе с описанием базовых принципов ее использования.

При изображении паттерна используется обозначение параметризованной кооперации языка UML (рис. 2.17) в виде пунктирного эллипса. В правый верхний угол эллипса встроено пунктирный прямоугольник, в котором перечислены параметры кооперации, представляющей тот или иной паттерн [22].

В последующем параметры паттерна могут быть заменены различными классами, чтобы получить реализацию паттерна в рамках конкретной кооперации. Эти параметры специфицируют используемые классы в форме ролей классов в рассматриваемой подсистеме.

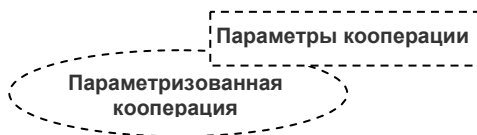


Рис. 2.17. Изображение паттерна в форме параметризованной кооперации

При связывании или реализации паттерна любая линия помечается именем параметра паттерна, которое является именем роли соответствующей ассоциации. В дополнение к диаграммам кооперации особенности реализации отдельных паттернов представляются с помощью диаграмм последовательности.

Паттерны проектирования позволяют решать различные задачи, с которыми постоянно сталкиваются проектировщики объектно-ориентированных приложений. В табл. 2.3 представлен полный список паттернов проектирования GoF и краткое описание назначения каждого из них. В качестве примера предлагается рассмотреть *паттерн «Фасад»*, предназначенный для замены нескольких разнотипных интерфейсов доступа к определенной подсистеме некоторым унифицированным интерфейсом, существенно упрощающим использование рассматриваемой подсистемы.

Общее представление паттерна проектирования «Фасад» может быть изображено с помощью диаграммы параметризованной кооперации (рис. 2.18) [22], которая содержит 4 параметра: класс Facade (Фасад), интерфейс IFacade, интерфейсы IConcreteClass и конкретные классы ConcreteClass, в которых реализованы интерфейсы IConcreteClass. Пунктирная линия со стрелкой в форме треугольника служит для обозначения отношения реализации.

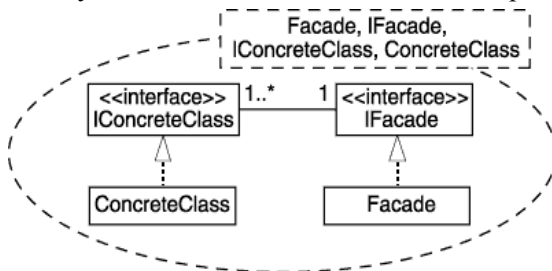


Рис. 2.18. Общее представление паттерна проектирования «Фасад»

Таблица 2.3

Полный список паттернов проектирования GoF

Название паттерна	Перевод	Назначение паттерна
1. Abstract Factory	Абстрактная фабрика	Предоставляет интерфейс для создания множества связанных между собой или независимых объектов, конкретные классы которых неизвестны
2. Adapter (синоним- Wrapper)	Адаптер (Обертка)	Преобразует существующий интерфейс класса в другой интерфейс, который понятен клиентам. При этом обеспечивает совместную работу классов, невозможную без данного паттерна из-за несовместимости интерфейсов
3. Bridge	Мост	Отделяет абстракцию класса от его реализации, благодаря чему появляется возможность независимо изменять и то и другое
4. Builder	Строитель	Отделяет создание сложного объекта от его представления, позволяя использовать один и тот же процесс разработки для создания различных представлений
5. Chain of Responsibility	Цепочка обязанностей	Позволяет избежать жесткой зависимости отправителя запроса от его получателя, при этом объекты-получатели связываются в цепочку, а запрос передается по цепочке, пока какой-то объект его не обработает
6. Command	Команда	Инкапсулирует запрос в виде объекта, обеспечивая параметризацию клиентов типом запроса, установление очередности запросов, протоколирование запросов и отмену выполнения операций
7. Composite	Компоновщик	Группирует объекты в иерархические структуры для представления отношений типа «часть – целое», что позволяет клиентам работать с единичными объектами так же, как с группами объектов

Продолжение табл. 2.3

Название паттерна	Перевод	Назначение паттерна
8. Decorator	Декоратор	Применяется для расширения имеющейся функциональности и является альтернативой порождению подклассов на основе динамического назначения объектам новых операций
9. Facade	Фасад	Предоставляет единый интерфейс к множеству операций или интерфейсов в системе на основе унифицированного интерфейса для облегчения работы с системой
10. Factory Method	Фабричный метод	Определяет интерфейс для разработки объектов, при этом объекты данного класса могут быть созданы его подклассами
11. Flyweight	Приспособленец	Использует принцип разделения для эффективной поддержки большого числа мелких объектов
12. Interpreter	Интерпретатор	Для заданного языка определяет представление его грамматики на основе интерпретатора предложений языка, использующего это представление
13. Iterator	Итератор	Дает возможность последовательно перебрать все элементы составного объекта, не раскрывая его внутреннего представления
14. Mediator	Посредник	Определяет объект, в котором инкапсулировано знание о том, как взаимодействуют объекты из некоторого множества. Способствует уменьшению числа связей между объектами, позволяя им работать без явных ссылок друг на друга и независимо изменять схему взаимодействия
15. Memento	Хранитель	Дает возможность получить и сохранить во внешней памяти внутреннее состояние объекта, чтобы позже объект можно было восстановить точно в таком же состоянии, не нарушая принципа инкапсуляции

Окончание табл. 2.3

Название паттерна	Перевод	Назначение паттерна
16. Observer	Наблюдатель	Специфицирует зависимость типа «один ко многим» между различными объектами, так что при изменении состояния одного объекта все зависящие от него получают извещение и автоматически обновляются
17. Prototype	Прототип	Описывает виды создаваемых объектов с помощью прототипа, что позволяет создавать новые объекты путем копирования этого прототипа
18. Proxy	Заместитель	Подменяет выбранный объект другим объектом для управления контролем доступа к исходному объекту
19. Singleton	Одиночка	Для выбранного класса обеспечивает выполнение требования единственности экземпляра и предоставления к нему полного доступа
20. State	Состояние	Позволяет выбранному объекту варьировать свое поведение при изменении внутреннего состояния. При этом создается впечатление, что изменился класс объекта
21. Strategy	Стратегия	Определяет множество алгоритмов, инкапсулируя их все и позволяя подставлять один вместо другого. При этом можно изменять алгоритм независимо от клиента, который им пользуется
22. Template Method	Шаблонный метод	Определяет структуру алгоритма, перераспределяя ответственность за некоторые его шаги на подклассы. При этом подклассы могут переопределить шаги алгоритма, не меняя его общей структуры
23. Visitor	Посетитель	Позволяет определить новую операцию, не меняя описаний классов, у объектов которых она вызывается

При решении конкретных задач проектирования паттерн «Фасад» может быть конкретизирован. В этом случае вместо параметров изображенной кооперации должны быть указаны классы, предназначенные для решения отдельных задач.

Паттерн «Фасад» может быть использован для выполнения операций по заданию и считыванию адресов сотрудников из базы данных. Фрагмент соответствующей диаграммы классов, изображенной на рис. 2.19 [22], содержит 2 класса: «Адрес» и «IАдрес» (интерфейс к операциям класса «Адрес»).

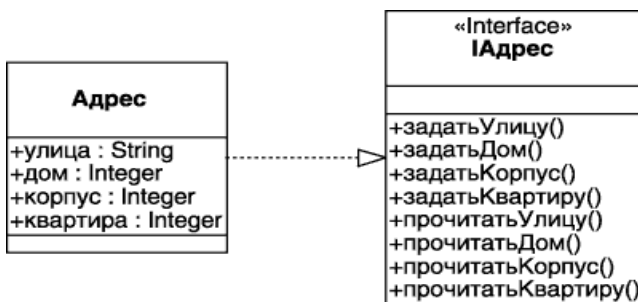


Рис. 2.19. Фрагмент диаграммы классов до применения паттерна «Фасад»

При задании адреса нового сотрудника необходимо обратиться к этому интерфейсу и последовательно выполнить операции: *зadатьУлицу()*, *зadатьДом()*, *зadатьКорпус()*, *зadатьКвартиру()*.

При этом в качестве аргумента используется идентификационный номер нового сотрудника. Для получения информации об адресе сотрудника необходимо также обратиться к этому интерфейсу и последовательно выполнить операции:

прочитатьУлицу(), *прочитатьДом()*, *прочитатьКорпус()*,
прочитатьКвартиру().

В качестве аргумента используется идентификационный номер интересующего сотрудника.

Отслеживать при каждом обращении правильность выполнения этой последовательности операций неудобно. С этой целью к данному фрагменту следует добавить еще один интерфейс —

реализацию паттерна «Фасад» для рассматриваемой ситуации. Соответствующий фрагмент модифицированной диаграммы классов будет содержать 4 класса (рис. 2.20), изображенные таким образом, чтобы иллюстрировать реализацию параметрической кооперации (рис. 2.21) [22].

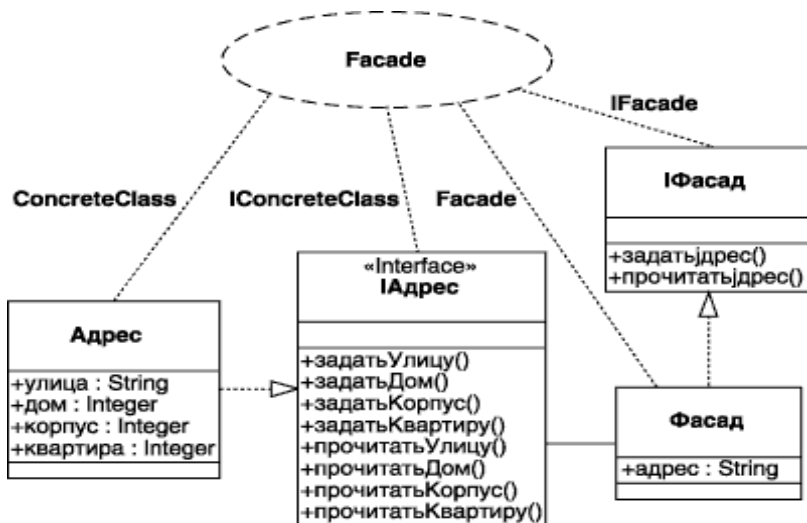


Рис. 2.20. Конкретная реализация паттерна проектирования «Фасад»

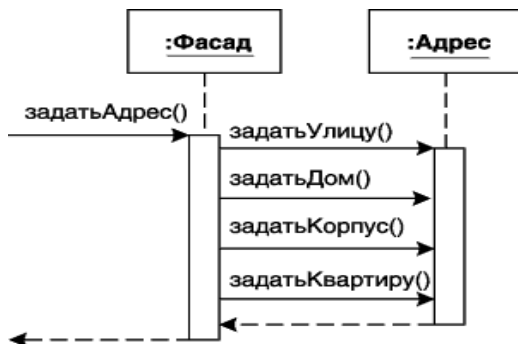


Рис. 2.21. Диаграмма последовательности для выполнения операции задания адреса

Диаграмма последовательности, представленная на рис. 2.21, может быть построена и для выполнения операции по чтению адреса. Использование паттерна «Фасад» обеспечивает для клиента не только простоту доступа к информации об адресах, но и независимость представления объектов класса «Адрес» от запросов клиентов. Это обстоятельство особенно актуально при изменении формата представления информации или смене соответствующей базы данных. В этом случае потребуется внести изменения только в реализацию операций класса «Фасад» [22].

В настоящее время паттерны проектирования реализованы в инструментальном средстве Model Maker 9 компании Model-Maker Tools BV [23], которое поддерживает нотацию языка UML и позволяет генерировать программный код на языке Delphi Pascal. Паттерны проектирования также реализованы в CASE-средстве Together 2008 компании Borland [24], которое поддерживает нотации языка UML и позволяет генерировать программный код на языке Java.

Важность паттернов для архитектуры предприятия в целом обусловлена следующими причинами [1]:

- если используются корректные паттерны, то вероятность получения адекватно работающей физической реализации архитектуры возрастает;
- разработка и использование паттернов в рамках предприятия в целом обеспечивает преимущества, связанные с их многократным использованием для решения различных проблем. Это дает архитекторам возможности по использованию опыта и стандартизации решений при создании новых систем;
- использование паттернов отделяет логический уровень от физического уровня архитектуры, что позволяет создать долговременно работающие решения и придает гибкость, поскольку на последующем этапе постоянные конструкции могут быть связаны с конкретными технологическими решениями.

В заключение следует отметить, что язык UML представляет собой нотацию для визуального моделирования программных систем и бизнес-процессов. В то же время описание языка UML не содержит сведений относительно того, каким образом и в ка-

кой последовательности следует разрабатывать канонические диаграммы при выполнении конкретных проектов. Соответствующая информация относится к области методологии проектирования программных систем. В настоящее время наиболее известны следующие методологии:

- 1) Rational Unified Process (RUP), разработанная и поддерживаемая компанией IBM Rational Software;
- 2) Microsoft Solutions Framework (MSF), разработанная и поддерживаемая компанией Microsoft;
- 3) Application Lifecycle Management (ALM), разработанная и поддерживаемая компанией Borland;
- 4) Extreme Programming (XP) — экстремальное программирование, поддерживаемое открытым сообществом независимых разработчиков.

Вопросы для самоконтроля

1. Дайте определение структурного анализа.
2. На каких общих принципах базируется методология структурного анализа?
3. Назовите структурные методологи построения архитектуры предприятия.
4. Опишите ключевые элементы диаграмм потоков данных DFD.
5. Опишите ключевые элементы диаграмм структурного анализа и проектирования SADT (IDEF0).
6. Из каких ключевых элементов состоят диаграммы и стандарт документирования процессов IDEF3?
7. Опишите диаграммы моделирования отношений между данными IDEF1X.
8. Назовите и охарактеризуйте основные принципы объектной модели.
9. Что собой представляет унифицированный язык моделирования UML?
10. Какой набор диаграмм входит в UML?
11. Для чего используют паттерны? Приведите их классификацию.

Глава 3

МЕТОДОЛОГИЯ ARIS ДЛЯ ПОСТРОЕНИЯ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ

3.1. Основы методологии ARIS¹

Моделирование реальных ситуаций в работе предприятий и построение комплексных бизнес-процессов с каждым годом становится все более актуальным. Появление множества различных методов моделирования приводит к значительным трудностям, связанным с их использованием в конкретных ситуациях. В связи с этим предпринимаются попытки создать стандартизованные концепции (архитектуры) для процесса разработки информационных систем и методов моделирования. Одной из таких концепций является «Архитектура интегрированных информационных систем» (ARIS — Architecture of Integrated Information Systems), разработанная А.-Е. Шеером. Эта концепция имеет два основных преимущества:

- 1) позволяет выбирать методы и интегрировать их, опираясь на основные особенности моделируемого объекта;
- 2) служит базой для управления сложными проектами, поскольку благодаря структурным элементам содержит встроенные модели процедур для разработки интегрированных информационных систем.

¹ Параграфы 3.1–3.7 третьей главы написаны с использованием материала «Инструментарий ARIS. Методы. Версия 4.1» [25].

Такая архитектура дает возможность вводить в применяемые методы элементы стандартизации. Концепция ARIS позволила создать комплексный метод моделирования бизнес-процессов, объединивший существующие и новые методы моделирования.

Архитектура ARIS явилась основой ARIS Toolset — инструментальной среды, разработанной компанией IDS Scheer AG. Инструментарий ARIS позволяет проводить построение, анализ и оценку рабочих процессов компании в терминах методологии организации бизнес-процессов. ARIS предоставляет достаточно простые средства для документирования и моделирования процессов.

В ARIS модель в целях упрощения делится на четыре отдельных типа, отражающих различные аспекты исследуемой системы (рис. 3.1):

1) **организационные модели**, представляющие структуру системы — иерархию организационных подразделений, должностей и конкретных лиц, связи между ними, а также территориальную привязку структурных подразделений;

2) **функциональные модели**, содержащие иерархию целей, стоящих перед аппаратом управления, и совокупность деревьев функций, необходимых для достижения поставленных целей;

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

4) **модели ресурсов**, представляющие используемые ресурсы. Эти модели важны для анализируемых бизнес-процессов только в той степени, в какой они необходимы для описания компонентов, более тесно связанных с самим бизнес-процессом. По этой причине компоненты других типов моделей (данные, функции и организационные структуры) рассматриваются с точки зрения их привязанности к ресурсам.

Типы моделей формируются таким образом, чтобы компоненты внутри каждого из них были тесно взаимосвязаны, в то время как отдельные типы в достаточной степени независимы. Декомпозиция бизнес-процесса на отдельные типы моделей уменьшает степень его сложности за счет исключения из рассмотрения

взаимосвязей между компонентами процесса, относящимися к другому типу моделей. В связи с этим вводится дополнительный тип модели — **управляющая модель**, в которой описываются взаимосвязи между моделями различных типов. Интеграция этих взаимосвязей с помощью модели специального типа позволяет вводить дополнительные взаимосвязи без какой-либо избыточности.

Управляющая модель — важнейший компонент архитектуры ARIS, отличающий ее от архитектур, предлагаемых другими авторами.

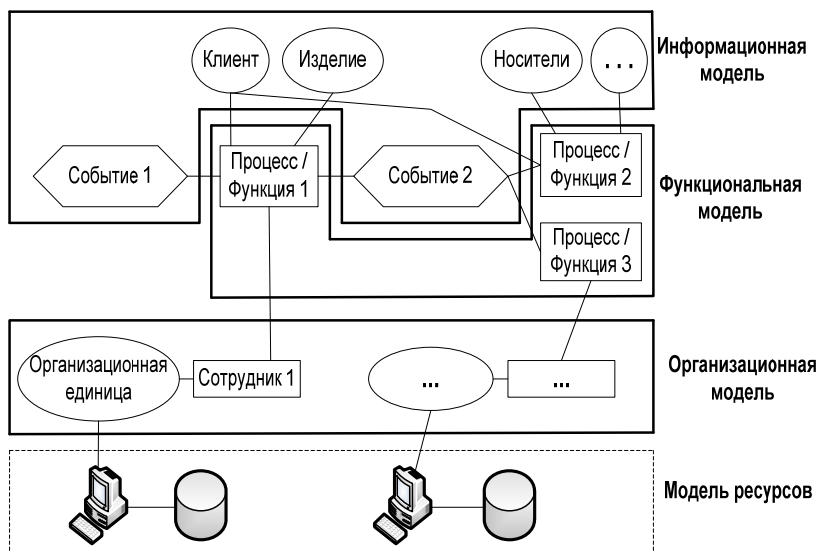


Рис. 3.1. Примеры моделей разных типов, составляющих описание процесса

Таким образом, для описания бизнес-процессов используются четыре модели, представленные на рис. 3.2. Создание различных типов моделей и проработка каждой модели по уровням описания в сочетании с формулировкой проблем бизнеса и составляет сущность процесса работы в архитектуре ARIS (рис. 3.3). Каждый тип модели подвергается разложению на три уровня: формулировку требований, спецификацию проекта и описание реализации.



Рис. 3.2. Модели архитектуры ARIS при описании бизнес-процесса

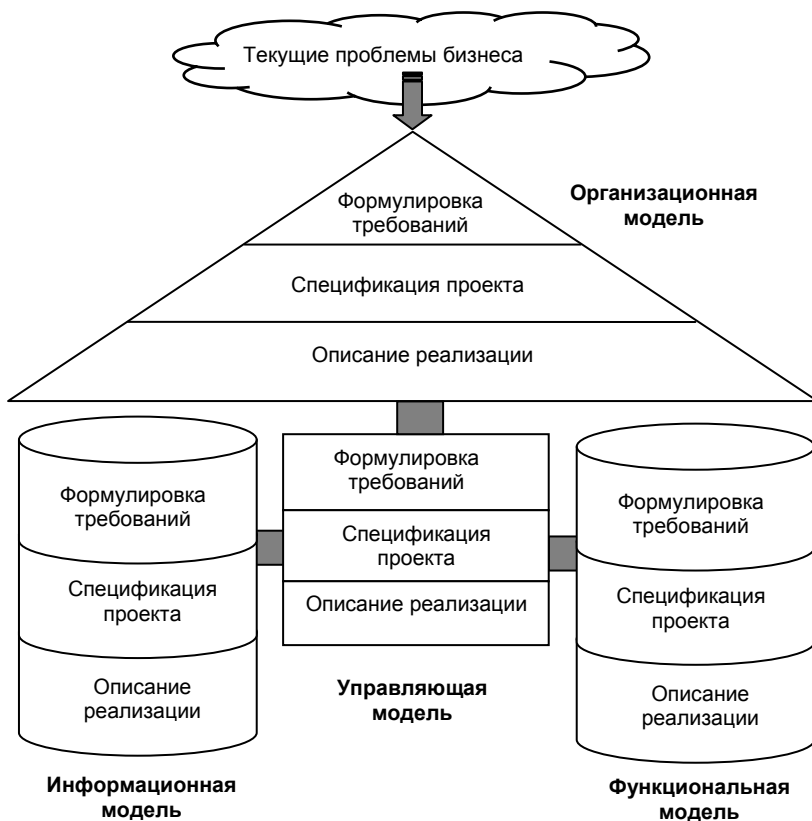


Рис. 3.3. Архитектура ARIS

Начальной точкой в разработке архитектуры предприятия, а затем и создании информационной системы является **анализ проблем бизнеса**. Модели на этом уровне — это описания бизнес-процессов с невысоким уровнем детализации. Однако они достаточно точно отражают цели, которые стоят перед пользователем информационной системы, и его язык. На этом этапе в описание включаются некоторые сведения по характеристикам будущей информационной системы, связанным с характеристиками бизнес-процессов. Для описания проблем бизнеса используются только полужформальные методы. Полученные модели еще не содержат детальной информации и однозначных технических формулировок, чтобы служить исходным материалом для их автоматической передачи непосредственно на этап реализации ИС.

На уровне формулировки требований для рассматриваемой проблемы бизнеса необходимо описать программное решение (прикладную ИС), которое должно поддерживаться формализованным описанием требований с целью последующего использования в качестве стартовой точки для трансляции сформулированных требований в программную систему. Этот процесс также очень близок к семантическому (смысловому) моделированию. Формулировка требований тесно связана с описанием проблем бизнеса.

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

Спецификация проекта может изменяться, не оказывая влияния на результаты предыдущего уровня формулировки требований. Но это не означает, что формулировка требований и спецификация проекта могут прорабатываться независимо друг от друга.

После завершения этапа формулировки требований его наиболее важная содержательная часть, отражающая категории управления бизнесом, должна быть определена таким образом, чтобы все относящееся к области информационных технологий и программных решений (например, производительность информационной системы), не влияло на предметное содержание.

На уровне описания реализации спецификация проекта трансформируется в конкретные аппаратные и программные компоненты. Так осуществляется физическая связь с ИС. Отдельные уровни описания имеют различные циклы корректировки. Частота корректировок выше всего на уровне описания реализации и ниже всего на уровне формулировки требований. Уровень описания реализации тесно связан с разработкой ИС: на этом уровне производится многократная корректировка функционирования системы по результатам коротких циклов (тестов) ее работы.

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

3.2. Организационная модель ARIS

Предприятие является сложной структурой, которая может быть разделена на легко управляемые элементы. Для упрощения работы при построении организационной модели определяются структурные схемы предприятия и устанавливаются правила поведения. Результат этого процесса — «правильная» организация.

Организационная структура включает правила, позволяющие произвести статичное структурирование предприятия.

Организационная схема — типичная форма представления организационных структур, описывающая организационные единицы и их взаимосвязи в зависимости от критериев, в соответствии с которыми организована структура.¹ **Организационные единицы** — это исполнители заданий, реализуемых для достижения целей деятельности предприятия. Вместе с организационными единицами в схему добавляются такие объекты, как **должности и конкретные фамилии сотрудников**. Организационным единицам может быть присвоен тип, определяющий их принадлежность к отделу или группе. Например, сотрудники могут соответствовать таким типам, как руководитель отдела, руководитель группы или менеджер проекта (рис. 3.4). С помощью типов объектов можно представить основные правила работы, которые являются абстрактным описанием конкретных организационных единиц. Например, в рамках некоторого процесса можно определить тип сотрудников, которым разрешено выполнение одной конкретной функции или доступ к отдельному информационному объекту.

Моделирование организационной структуры компании — стартовая точка в создании топологии компьютерной сети, которая должна быть определена на уровне спецификации проекта и которая, как предполагается, будет поддерживать организационную структуру наиболее оптимальным образом. Соединения сети и сетевые узлы, расположенные в определенных местах компании, являются главными элементами топологии сети. Таким образом, местоположение организационной единицы — это наиболее важная связь между сформулированными требованиями и спецификацией проекта. Следовательно, для каждой организационной единицы можно определить место, причем это должно быть сделано как можно раньше — на уровне формулировки требований (рис. 3.5).

¹ В диаграммах методологии ARIS, приведенных в данной главе, отсутствует свойство объектов «цвет», но это не ухудшает восприятие, поскольку объекты имеют разную форму изображения.

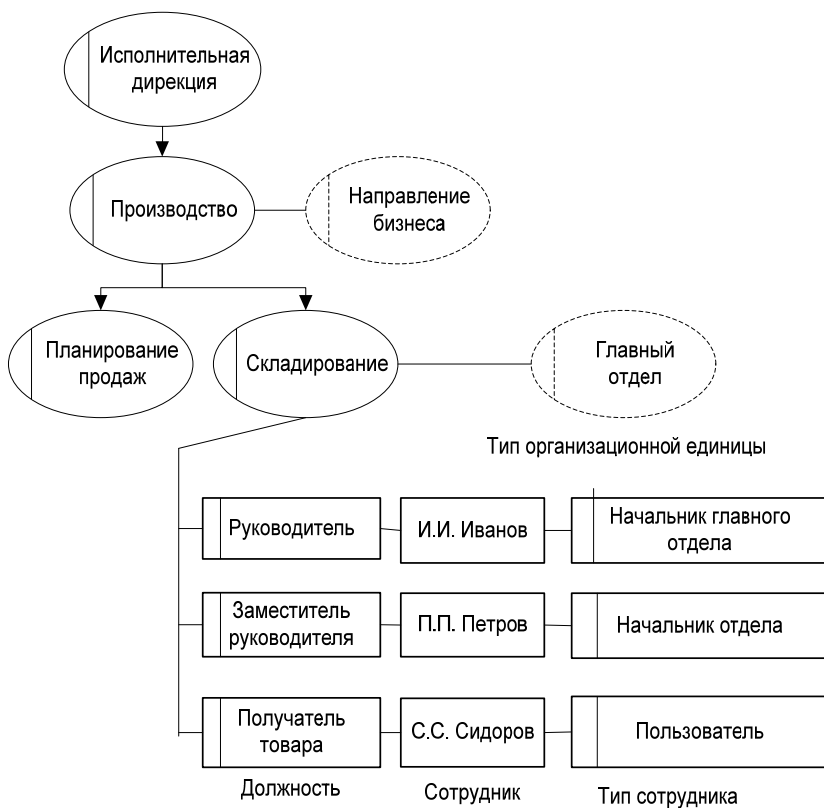


Рис. 3.4. Организационная схема должностей и сотрудников

Местоположение может занимать любую позицию в иерархической структуре организации и определять как отдельное здание, так и, при более детальном анализе, отдельный офис или даже единственное рабочее место.

Таким образом, на этапе спецификации проекта возможно соотнести узлы сети с отдельным рабочим местом в организационной единице.

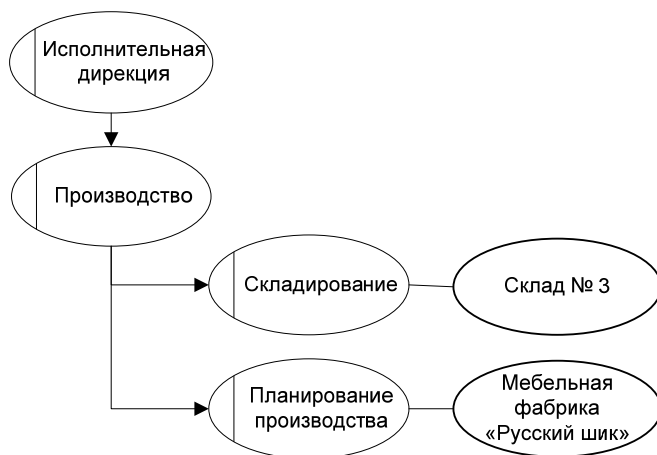


Рис. 3.5. Описание местоположений

Организационная диаграмма ARIS поддерживает также *календарь смен* как многоуровневую объектную модель:

- на нижнем уровне находятся объекты типа «Перерыв». *Перерыв* — это ежедневный интервал времени в рамках смены, в течение которого не выполняется никакая работа;

- следующий уровень в иерархии включает объекты типа «Смена». *Смена* — это ежедневный интервал времени, в течение которого выполняется работа. Смена определяется относительным временем начала и продолжительностью. Смена может иметь один и более перерывов. Относительное время начала всех перерывов должно находиться во временных рамках смены;

- *цикл смен* — еженедельный интервал времени или интервал продолжительностью 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 должен повторяться непрерывно, для идентификации этого вводится специальный атрибут «Период», который определяет, как часто данный цикл должен повторяться;

- *план смен* — это совокупность всех циклов и их смен, которая описывает рабочие часы некоторого ресурса. Описание содержит периодически повторяющиеся объекты. В соответствии со специальными правилами определяются объекты (отпуск, болезнь, праздник и другие дни, в течение которых работа не выполняется), которые не включаются в описание.

Организационная структура предприятия, представленная организационной схемой, может поддерживаться коммуникационной и информационной инфраструктурами системы. Структурные требования к таким информационным системам в основном могут быть описаны на этапе спецификации проекта с использованием топологии сети, для этого используются специальные диаграммы сети. На диаграмме сети устанавливаются *два типа связи*: связь описания типа сети со спецификацией проекта и связь описания компонентов сети и конкретных местоположений с формулировкой требований.

3.3. Функциональная модель ARIS

Функциональная модель ARIS отражает функции, выполняемые на предприятии, и средства из других типов моделей, которые обозначают связи между функциями. **Функция** — это предметно-ориентированное задание или действие, выполняемое над объектом, в результате которого достигается одна или несколько целей, стоящих перед компанией. Функции отображаются в виде прямоугольников с закругленными углами.

Функции могут быть представлены с различными уровнями детализации. На *верхнем уровне* описываются наиболее *сложные функции*, характеризующие отдельный бизнес-процесс или последовательность процессов, например, процесс обработки запроса клиента на всем его протяжении, начиная от получения запроса клиента до отгрузки товара. Такой процесс состоит из сложной функции, которая может быть разделена на подфункции. Следовательно, термин «функция» может быть использован на

всех уровнях детализации. Последовательная детализация функций образует иерархическую структуру их описаний. Для более содержательного описания отдельного уровня иерархии могут быть использованы также другие термины: «транзакция», «процесс», «подфункция», «базовая функция» («операция»).

Разделение функций на элементы может происходить на нескольких иерархических уровнях. *Базовые функции* представляют *нижний уровень* в семантическом дереве функций. К базовым относятся функции, которые уже нельзя разделить на составные элементы с целью анализа бизнес-процесса.

Для представления иерархической структуры функций служит **диаграмма дерева функции**, или **иерархическая диаграмма**. 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). На рис. 3.6. изображен пример процессно-ориентированного функционального дерева.

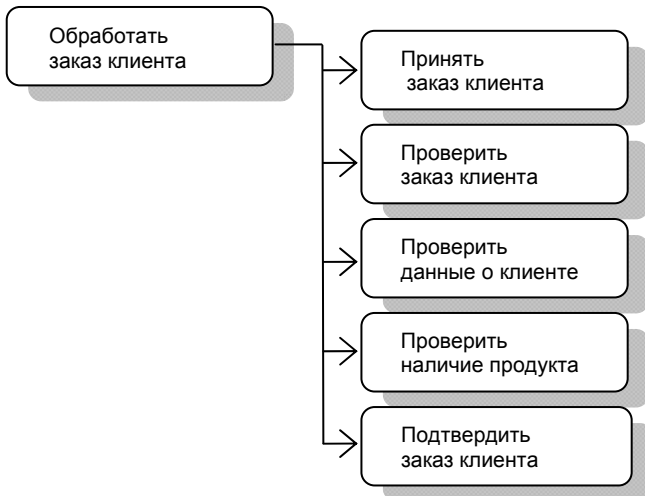


Рис. 3.6. Процессно-ориентированное функциональное дерево

Способ представления в виде функционального дерева позволяет уменьшить степень сложности и является статичным описанием функции. Динамические описания могут потребоваться при анализе последовательности функций, выполняемых в хронологическом порядке в рамках некоторой процедуры.

При описании функции с объектно-ориентированной точки зрения используется не только такое ее свойство, как возможность декомпозиции на элементы, но и другие свойства функции, представляющие интерес. В особенности это относится к свойствам, которые учитываются при проектировании бизнес-процессов.

Таким образом, каждое описание функции должно включать информацию о том, будет ли эта функция инициирована пользователем или она может работать автоматически. Это позволяет объединять все аналогичные функции, не требующие вмешательства пользователя, в один пакет (пакетное задание).

При реорганизации бизнес-процессов 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 как атрибуты объекта типа «Функция».

Одной из диаграмм, используемых для описания функций, является **Y-диаграмма** (рис. 3.7).

Y-диаграмма представляет функции (задания) компании на верхнем уровне агрегации. Здесь участвуют основные макрофункции: прототипирование изделия, управление материалами, обслуживанием. Левая ветвь диаграммы содержит основные управленческо-административные функции, связанные с планированием и управлением производством, а правая — технико-ориентированные функции планирования производства и реализации продукции.

Функции планирования расположены в верхних частях Y, а функции управления и реализации — в нижней части.



Рис. 3.7. Y-диаграмма

Диаграмма SAP-приложений позволяет представлять модели-прототипы SAP R/3¹, ориентированные на модули системы управления предприятием SAP R/3. В модели-прототипе R/3 матрица выбора процессов связана с каждым объектом диаграммы данного типа. Она отображает основные процессы, доступные в отдельных модулях R/3, и сценарии процессов.

Прежде чем начать моделирование, анализ или оптимизацию рабочего процесса, необходимо определить цели компании

¹ SAP R/3 — широкоизвестная интегрированная автоматизированная система управления предприятием, ориентированная на крупные и средние предприятия.

в области совершенствования бизнес-процессов. Для задания целей используется диаграмма целей, с помощью которой можно также построить иерархию целей (рис. 3.8). Данный тип диаграмм связывается с другими диаграммами на уровне формулировки требований с помощью объекта типа «Функция». Для каждой цели можно отобразить функцию (бизнес-процесс), которая ведет к достижению цели. При моделировании и оптимизации бизнес-процессов необходимо указать приоритеты объектов и соответствующих функций.

Уровень спецификации проекта для функциональной модели 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.

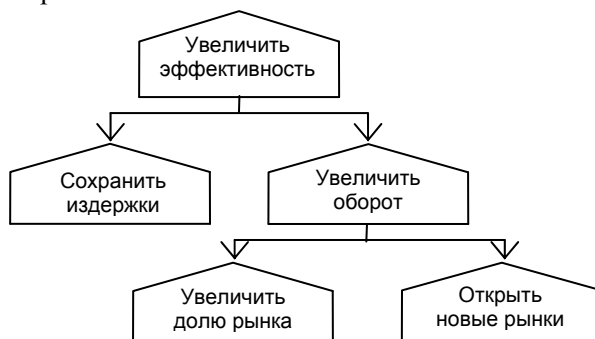


Рис. 3.8. Пример диаграммы целей

На уровне спецификации проекта в рамках функциональной модели необходимо ответить на следующие ключевые вопросы:

- какой может быть поддержка функций, определенных с помощью типов ПС, типов модулей или проектов этих функций;
- можно ли что-либо сказать о модульной структуре ПС или типах модулей;
- какие списки и экраны потребуются для выполнения функции;

- какие списки могут быть созданы с помощью прикладной системы данного типа или модуля данного типа и какие экраны поддерживают прикладную систему и модули данных типов;
- какая технологическая база имеется в распоряжении для реализации прикладной системы данного типа (операционная система, интерфейс пользователя или система управления БД);
- как соотносится с целями компании прикладная система определенного типа.

3.4. Информационная модель ARIS

Формулировка требований в рамках модели данных включает описание семантической модели данных в рассматриваемой предметной области. В соответствии с принципом разделения ARIS это описание содержит и объекты, специфицирующие начальное и конечное события в цепочке процесса, и описание состояний инфраструктуры, связанной с процессом. При сравнении методов моделирования функций и данных к последним предъявляются особые требования с точки зрения применяемого метода. В функциональной модели единственный рассматриваемый объект — это функция. В терминах взаимосвязей между функциями описываются только отношения старшинства и подчиненности.

Наиболее распространенным методом создания семантических моделей является использование модели Чена «сущность-отношение» (ERM). Этот метод моделирования оперирует такими терминами, как тип сущности, атрибут и др. Многочисленные взаимосвязи между этими объектами значительно сложнее, чем в функциональном моделировании. Детальная информация по данному вопросу содержится во многих работах по проектированию баз данных.

В течение последних лет оригинальная модель Чена была значительно расширена, что достаточно существенно для модели данных в архитектуре ARIS.

Операторы проектирования обеспечивают формальную поддержку процесса создания модели данных. Их применение гарантирует системность подхода и дает возможность тому, кто знает существующую структуру данных, понять суть процесса проектирования. С помощью операторов проектирования разработаны новые концепции, основанные на существующих. Процесс проектирования является интеллектуальной процедурой, в большей степени выполняемой на уровне управления корпоративными знаниями. Анализ условий выполнения бизнес-процессов с точки зрения их структур данных помогает разработчикам структурировать известные условия, базируясь на новом представлении, и создавать новые отношения, не рассмотренные до сих пор.

С учетом различных подходов к расширению модели «сущность-отношение» (eERM) выделяется четыре основных оператора проектирования:

1) **классификация**. При помощи данного оператора объекты (сущности) одного и того же типа идентифицируются и ассоциируются в соответствии с некоторым признаком (типом сущностей). Один объект идентичен другому, если он описан теми же свойствами (атрибутами);

2) **обобщение**, посредством которого аналогичные типы объектов группируются под одним из старших типов объекта;

3) **агрегация**. С помощью этого оператора описывается формирование нового типа объекта с помощью комбинации существующих типов объектов. В данном контексте новый тип объекта может нести новые свойства;

4) **группировка**, в процессе которой формируются группы, составленные из элементов некоторого множества сущностей.

Для моделирования, особенно моделирования данных, характерно наличие множества терминов, определяющих информационные объекты. Например, термин «заказ» в отделе закупок полностью отличается от понятия заказа у сотрудников производственного отдела. При введении соответствующей единой терминологии для предприятия и его отделов определяемая информация становится более понятной. По этой причине набор методов ARIS содержит так называемые **модели технических**

терминов, которые не только позволяют манипулировать различными терминами как синонимами, но и дают возможность поддерживать отношения между объектами в моделях данных (тип сущности, тип отношения и т. д.).

Для представления этих отношений вводится тип объекта *«Технический термин»*. Теперь с каждым информационным объектом модели данных могут быть связаны разные технические термины (рис. 3.9), для которых устанавливаются взаимосвязь и иерархическая упорядоченность.

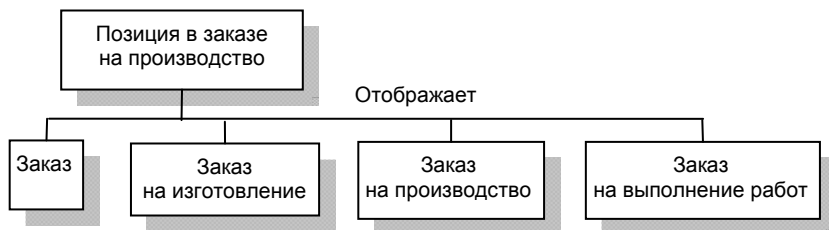


Рис. 3.9. Технические термины

Термины, определяемые моделью технических терминов, могут использоваться и в других диаграммах, которые содержат информационные объекты, например в диаграммах процессов для представления входа/выхода данных для функции.

С помощью **диаграмм атрибутов eERM** можно описать атрибуты для каждого типа сущности и отношения на отдельной диаграмме. В эту диаграмму можно включить тип объекта из диаграммы eERM (тип сущности или тип отношения) в виде копии экземпляра. Таким образом может быть смоделировано распределение атрибутов по объектам eERM. В этом контексте можно определить, является ли атрибут, связанный с объектом eERM, ключевым атрибутом, внешним ключом или описательным атрибутом. На рис. 3.10 приведен соответствующий пример.

Кроме представления и распределения отдельных атрибутов eERM, на этом типе диаграммы можно также отобразить группу типов атрибутов и их распределение. *Группа типов атрибутов*

представляет группу атрибутов одного типа сущности из диаграммы eERM, которые семантически близко связаны. Это позволяет создать группу атрибутов, содержащую все атрибуты eERM, которые вместе образуют, например, вторичный ключ.

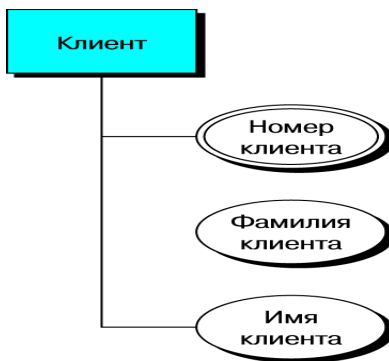


Рис. 3.10. Атрибуты eERM для типа сущности

Реляционная диаграмма и диаграмма атрибутов используются для определения существующих отношений, атрибутов и их взаимосвязей с информационными объектами, вводимыми при формулировке требований. При формировании реляционной диаграммы могут быть определены необходимые отношения. **Отношение описывает тип сущности по ее атрибутам.** Это подмножество всех возможных комбинаций диапазонов значений отдельных атрибутов.

Каждая сущность, входящая в модель ER, представляет собой отношение в реляционной модели. При преобразовании типов отношений модели eER важным аспектом в решении вопроса, должно ли создаваться соответствующее отношение для каждого отдельного типа отношений, является мощность отношений.

Отношения «многие ко многим» ($n:m$), отличные от отношения «один ко многим» ($1:n$), должны быть представлены соответствующими отношениями. Для каждого отдельного отношения реляционная диаграмма указывает, какой тип сущности или отношения модели eER представлен.

Отношение может быть в дальнейшем специфицировано перечнем его атрибутов. В зависимости от вида определенного атрибута (ключевой, внешний ключевой или описательный) он может быть обозначен с помощью соединения, связывающего отношение и его атрибут. С другой стороны, на основе сформулированных требований можно установить отношение каждого отдельного атрибута к атрибуту модели ER (рис. 3.11).



Рис. 3.11. Представление атрибутов и объектов данных на уровне формулировки требований

Для уменьшения степени сложности представления атрибуты каждого отношения могут быть описаны с помощью связанной с данным отношением диаграммы атрибутов (рис. 3.12).



Рис. 3.12. Диаграмма атрибутов

На уровне спецификации проекта кластеры данных, содержащие сформулированные требования, описываются с помощью типа объекта «Представление». Представление определяется на основе определения кластера данных. Представление понимается как логический взгляд на некоторую совокупность отношений. Отношения, связанные с Представлением, также могут быть описаны реляционной диаграммой (рис. 3.13).

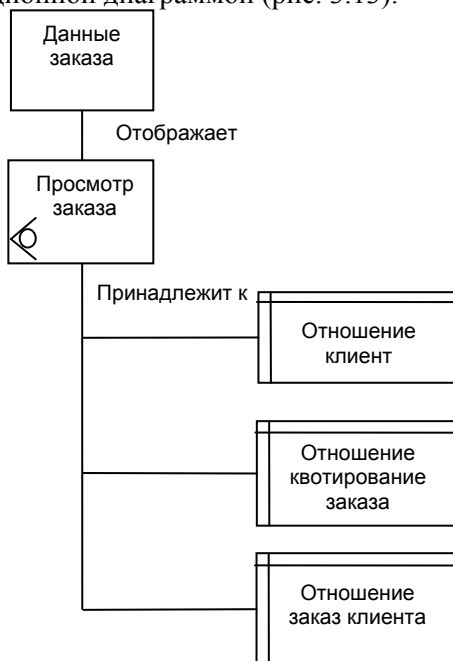


Рис. 3.13. Определение объекта «Представление»

Отношения мощностью «1:n» отражаются интеграцией ключевых атрибутов старших типов сущностей с отношением подчиненных типов сущностей. При этом первоначальный ключевой атрибут становится внешним ключом отношения.

Атрибут в реляционной модели, отражающий тип отношения в модели ER, может быть представлен в реляционной диаграмме обычным соединением (рис. 3.14).

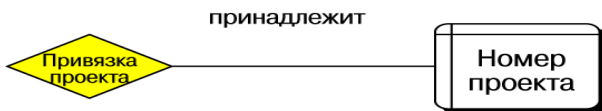


Рис. 3.14. Связь атрибута и типа отношения в модели ERM

Помимо диаграмм ERM-атрибутов методология ARIS поддерживает диаграммы системных атрибутов. В отличие от ERM-атрибутов, основное свойство системных атрибутов состоит в представлении и управлении данными, ориентированными на интерфейс *ARIS Toolset*. Системные атрибуты объектов имеют два поля значений, которые заполняются соответствующей информацией. Это гарантирует большую гибкость в экспортируемом содержимом.

Таблицы и поля систем управления БД могут быть описаны табличной диаграммой, к каждой таблице — «привязаны» поля. При дальнейшей спецификации каждому полю присваиваются индексы сортировки и области их изменений (рис. 3.15).

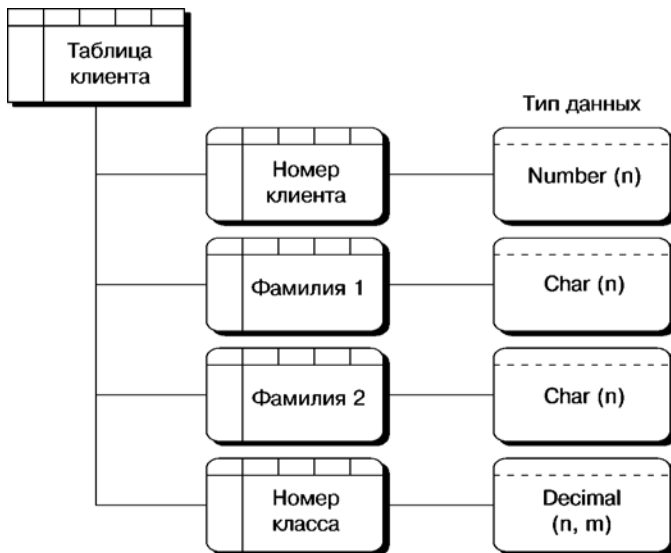


Рис. 3.15. Размещение полей

Преобразование и документирование таблиц и полей БД, используемых в компании, иногда производится без учета определений реляционной схемы. Именно поэтому отношения реализации могут быть отображены не только между отношениями (или атрибутами) и таблицами (или полями), но и между типами сущностей (или атрибутами модели ER) и таблицами (или полями).

Можно также показать, какие отношения и атрибуты выполняются или (оставляя в стороне реляционные определения) какие типы сущностей, типы отношений и ERM-атрибуты отображаются таблицами и полями. На рис. 3.16 приведен пример этих двух форм представления.

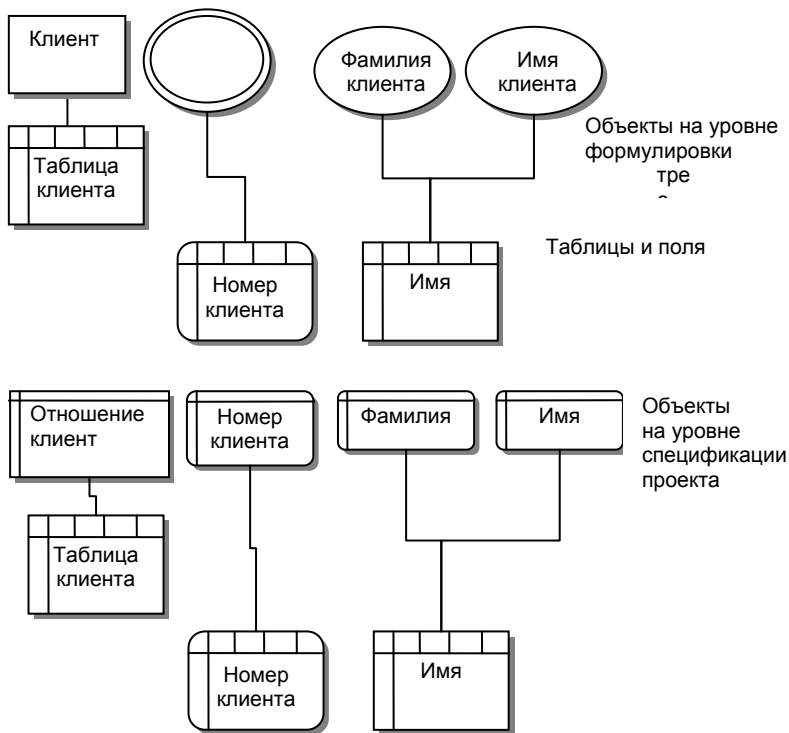


Рис. 3.16. Описание объектов на уровне формулировки требований и спецификации проекта

Для установления точного местоположения некоторых таблиц и полей в компании необходимо определить каждый отдельный экземпляр таблицы. То же самое относится к случаю, когда предполагается, что организационные единицы имеют авторизованный доступ к определенным таблицам и полям. Тип объекта «Таблица», введенный ранее, определяет логическую структуру физической таблицы и ее поля на уровне типа. Определенные таким образом многочисленные экземпляры каждой таблицы могут быть доступны (сохранены в другой среде) в различных местах компании. Для иллюстрации этого факта вводятся типы объектов «Таблица» (образец) и «Поле» (образец). Благодаря этим объектам можно точно определить количество экземпляров таблиц и полей. Эти связи представлены на рис. 3.17.

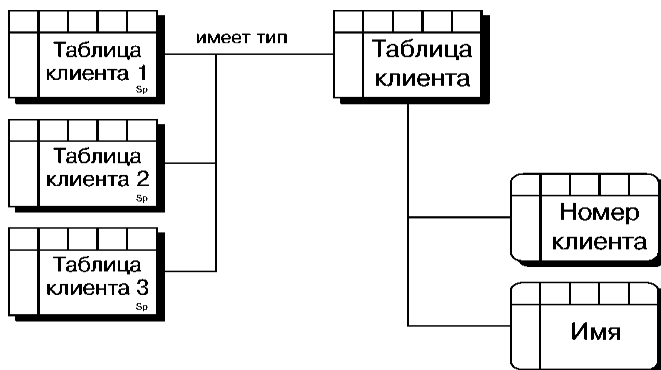


Рис. 3.17. Экземпляр таблицы

3.5. Управляющая модель ARIS

Перед моделированием отдельных компонентов архитектуры ARIS необходимо осмыслить содержательную часть бизнес-процесса, т. е. понять проблемы бизнеса. Недостатки используемых ИС должны быть представлены в формулировках, позволяющих концепцию проектируемой системы ориентировать на поддержку бизнес-процессов и целевые установки бизнеса.

Выявленные недостатки позволяют более четко определить цели, которые должны быть достигнуты в результате вновь создаваемых ИС. Разработчики ARIS утверждают, что требования к полноте описания текущего состояния бизнеса, а также компактности отображения недостатков имеющихся систем обуславливают ограниченность возможностей общеизвестных методов моделирования. Основные возможности этих методов рассчитаны на анализ различных аспектов бизнеса, поэтому они могут применяться только для создания отдельных типов моделей. Взаимосвязи процессов представляются в компактном виде с помощью диаграмм (PCDs), которые также позволяют описывать ИС.

Диаграмма типа PCDs (Process Chain Diagram) — это диаграмма цепочки процесса (диаграмма процесса). В диаграмме этого типа цепочка процесса отображается в виде замкнутого цикла. Отдельные типы моделей бизнес-процессов, определяемые ранее (организационная модель, модель данных, функциональная модель и модель ресурсов), а также их взаимосвязи выражены более отчетливо. Пример диаграммы процесса изображен на рис. 3.18. Столбцы «Событие» и «Функция» отображают логическую последовательность выполнения процесса. Отдельные функции процесса представлены в столбце «Функция». Они связаны с событиями, инициирующими выполнение функций или переключение между ними. Функции и события связаны пунктирными стрелками, указывающими события, которые переключают функции, а также события, которые сами генерируются функциями. Таким образом, эти стрелки отображают поток управления функциями.

В приведенном примере функция «Ввести заказ» инициируется событием «Заказ получен». В результате выполнения этой функции происходит событие «Заказ введен», которое, в свою очередь, инициирует следующую функцию «Обработать заказ». Это отношение между событиями и функциями составляет процедурную последовательность функций как логическую цепочку событий, которую называют цепочкой процесса.

Логическая взаимозависимость возможных точек ветвления и циклов потока управления может быть выражена посредством логических операций, связывающих функции и события.

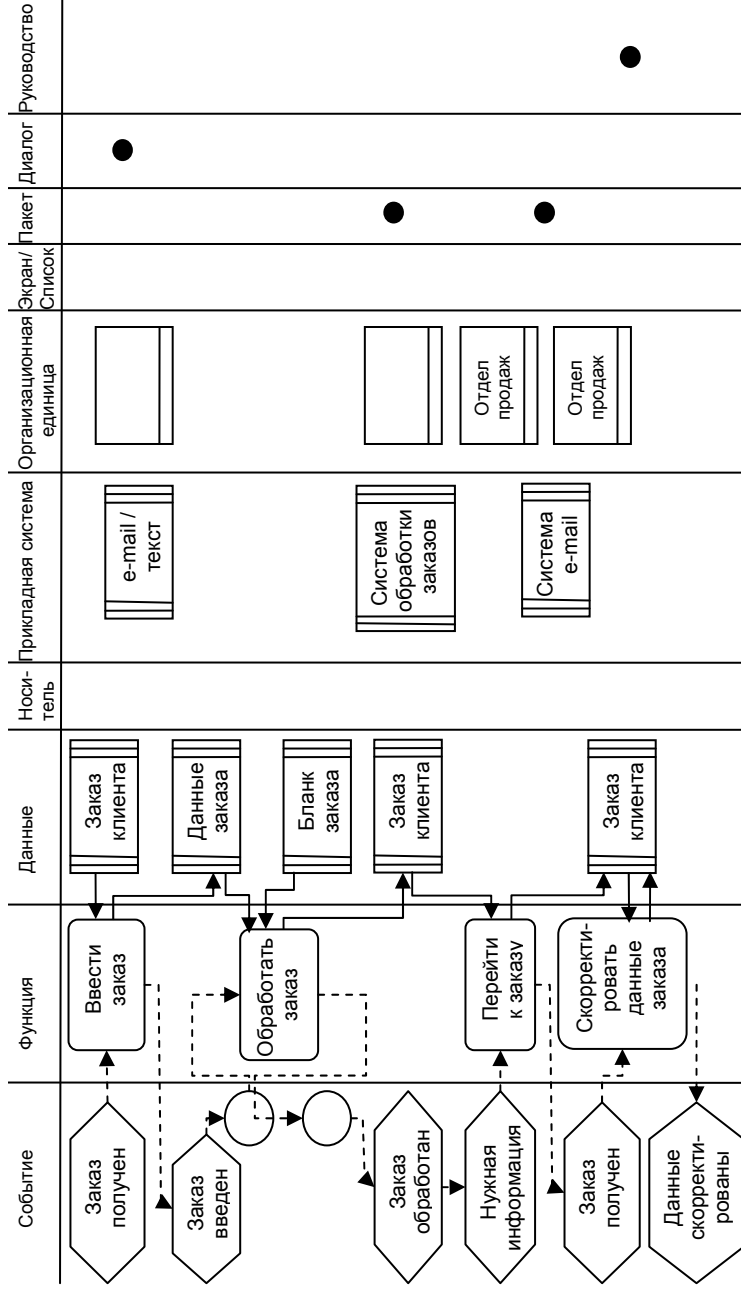


Рис. 3.18. Диаграмма процесса «Обработка заказа»

Входные и выходные данные, требующиеся функциям, показаны в столбце «Данные» в виде кластеров данных. Входные данные для функции «Обработать заказ» — это данные заказа; генерируемые выходные данные — заказ клиента. Отображаться могут не только информационные объекты, но и носители (среда), на которых находится информация. Это могут быть документ, список, написанная вручную квитанция или устройство памяти (например, жесткий диск).

Организационные единицы (отделы), ответственные за выполнение каких-либо отдельных функций, показаны в столбце «Руководство».

Столбцы «Диалог», «Пакет», «Руководство», которые характеризуют тип обработки, и столбец «Прикладная система» представляют дополнительную информацию о степени использования информационных технологий для поддержки отдельной функции.

При анализе бизнес-процесса, отображающего некоторую реальную ситуацию, недостатки его организации или причины неэффективности могут быть определены с помощью диаграммы процесса. Такими недостатками могут быть дезинтеграция между ручной обработкой и обработкой с помощью информационных технологий или организационная дезинтеграция (например, частая смена сотрудников отдела, выполняющих данную функцию). Кроме того, лишние входы (процедурная избыточность данных) и задержки в выполнении функций становятся четко видимыми. Это приводит к появлению множества идей по совершенствованию данного бизнес-процесса.

При описании начальной ситуации диаграммы процессов создаются с относительно высоким уровнем обобщения (без детализации). Поскольку эти диаграммы используются в основном для отображения взаимосвязей всех компонентов ARIS (моделей различных типов), они служат основой для создания управляющей модели ARIS. При ее создании используются не только диаграммы процесса, но и диаграммы цепочки процесса *EPCs (Event-driven Process Chain)*, управляемого событиями. Такие диаграммы называют событийными диаграммами процессов.

С точки зрения моделирования событийные диаграммы процессов также полезны, как и простые диаграммы процессов. Единственное отличие состоит в том, что элементы диаграмм EPCs не должны располагаться в предопределенных столбцах. Таким образом, модель небольшой процедуры может быть построена с помощью одного из методов (PCDs или EPCs), в то время как для представления модели всего бизнес-процесса предпочтительнее EPCs.

Диаграмма типа EPCs — это событийная диаграмма процесса, отображающая упорядоченную последовательность комбинаций событий и функций. С помощью диаграмм типа EPCs процедуры бизнес-процесса представляются как логические последовательности событий.

Под событием в ARIS понимают факт получения информационным объектом связанного с бизнес-процессом статуса, который управляет или воздействует на дальнейшее выполнение бизнес-процесса. События переключают функции, т. е. передают управление от одной функции к другой; они также могут быть результатом выполнения функций. В отличие от функций, имеющих некоторую продолжительность, события происходят моментально.

События графически изображаются в виде шестиугольников (рис. 3.19). Описание события должно содержать не только информационный объект («заказ»), но и описание изменения состояния («получен»). Поскольку события определяют, какое состояние (или отношение) будет переключать функцию и какое состояние будет определять конец ее выполнения, начальные и конечные узлы EPC всегда являются событиями.

Одно событие может инициировать выполнение одновременно нескольких функций, и наоборот, функция может быть результатом наступления нескольких событий. Эти ветвления и циклы обработки отображаются на диаграмме EPC с помощью соединителей в виде небольшого кружка. Соединители могут не только отображать графические связи между элементами модели, но и определять логические связи между объектами.

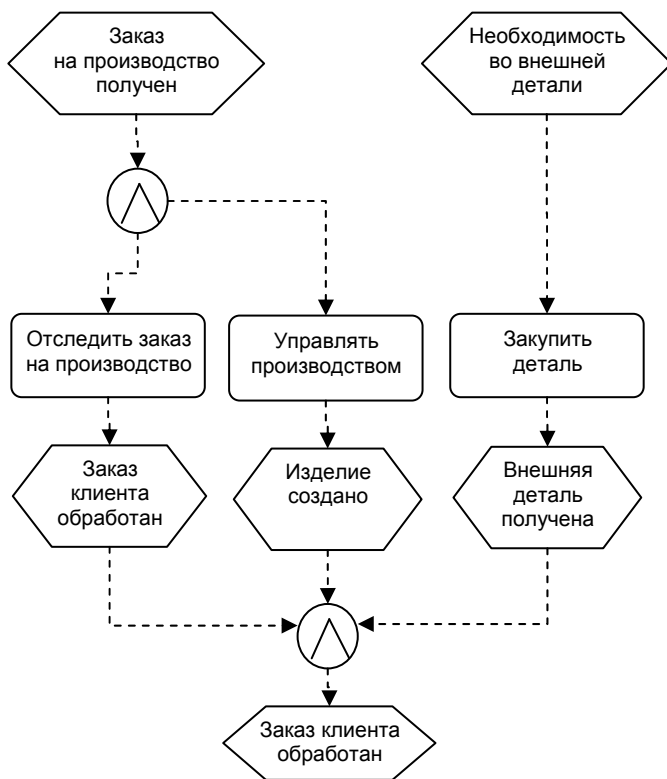


Рис. 3.19. Пример диаграммы EPC

Преобразование входных данных в выходные может быть отражено диаграммами описания функции (вход/выход), которые в основном соответствуют обычным диаграммам входа/выхода, используемым в других методах.

Диаграммы описания функций (вход/выход) содержат функции из функциональной модели и информационные объекты из модели данных. Стрелки определяют, используются ли информационные объекты в качестве входных данных, выходных данных или входных/выходных данных.

В диаграммах PCD объекты должны быть упорядочены в колонки. В диаграмме EPC допустима свободная организация объектов. Однако добавление входных/выходных данных может привести к путанице в моделях, поэтому в диаграммах PCD следует придерживаться последовательности в представлении, соответствующей последовательности выполнения функций в бизнес-процессе. На рис. 3.20 диаграмма EPC с входными/выходными данными представлена как диаграмма PCD.

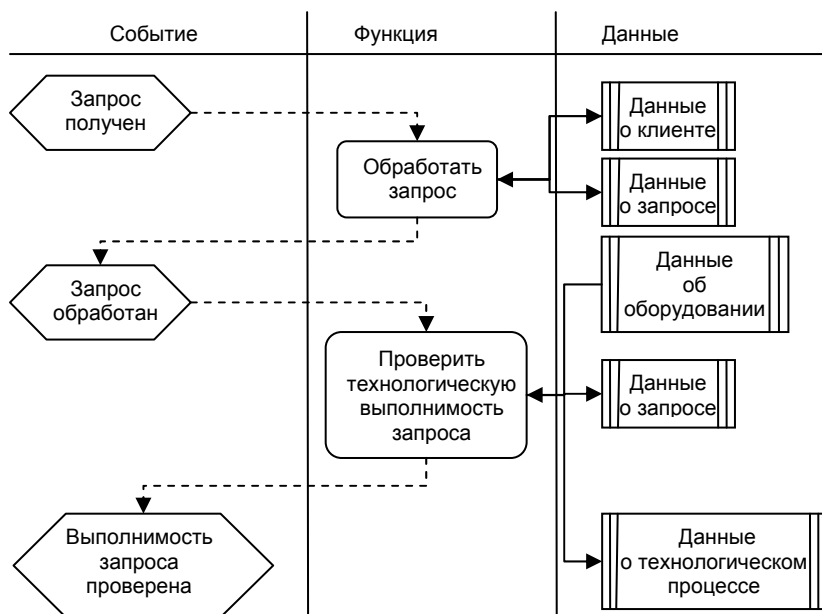


Рис. 3.20. Диаграмма PCD с данными входа/выхода

Для отображения потока данных между функциями можно использовать диаграммы информационных потоков (рис. 3.21).

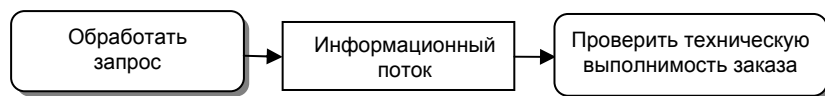


Рис. 3.21. Диаграмма информационных потоков

Отражение информационных объектов описывается событиями. С каждым событием связываются конкретные информационные объекты модели данных, причем событие определяет состояние (статус) этих информационных объектов на заданный момент времени.

Вначале события описываются в общем виде на верхнем уровне выполняемого процесса по методу «сверху вниз», следующий шаг при моделировании процесса заключается в детальной спецификации событий. Если они упорядочены каким-либо способом, наступление событий фиксируется на верхнем уровне описания. С помощью диаграммы событий можно представить указанную связь событий на верхнем и более детальном уровнях моделирования (рис. 3.22).

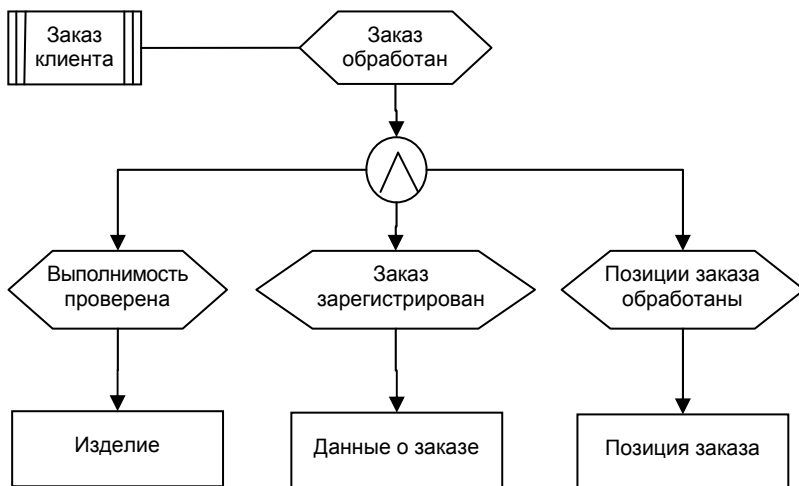


Рис. 3.22. Диаграмма событий

Для лучшего понимания связи функции с организационными единицами и данными объекты «организационные единицы» также вводятся на диаграммы PCD и eEPC (рис. 3.23–3.24).

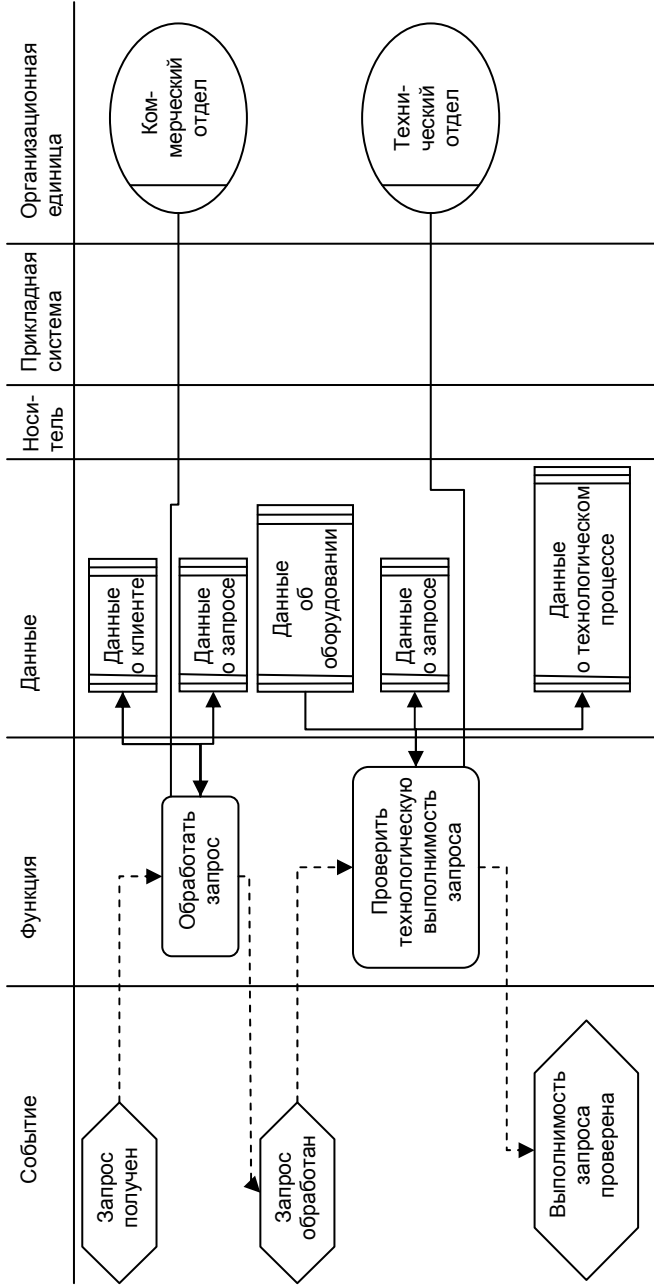


Рис. 3.23. Диаграмма PCD с учетом организационных единиц

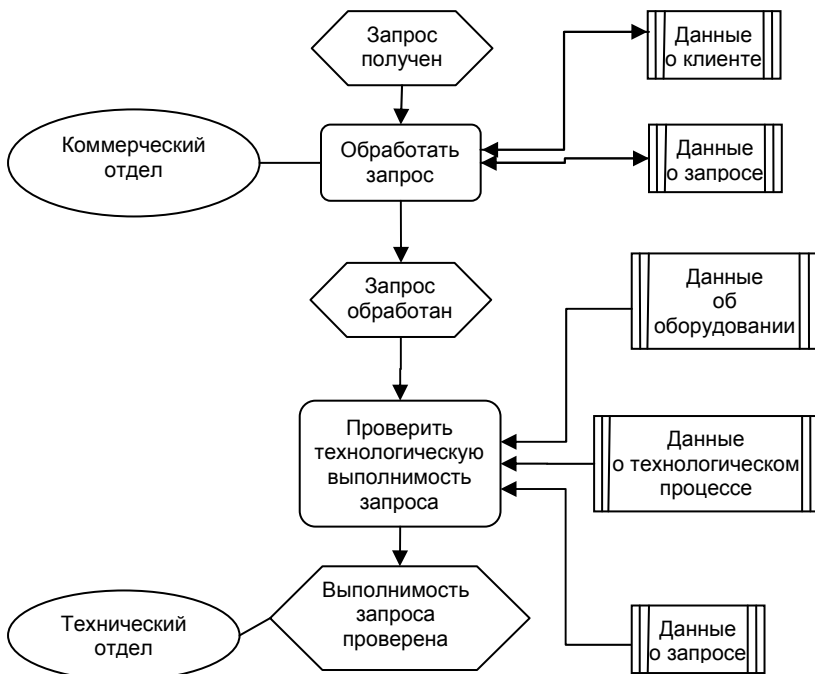


Рис. 3.24. Диаграмма ePC с учетом организационных единиц

В методологии ARIS при построении управляющей модели могут использоваться:

- **диаграммы цепочки добавленного качества**, позволяющие отобразить не только старшинство и подчиненность функций, но и их связи с организационными единицами и информационными объектами;
- **диаграммы правил**. В цепочке процесса можно использовать правила в виде операторов для спецификации тех операторов, которые связывают события и функции;
- **диаграммы коммуникаций**. Большие модели-прототипы содержат огромное разнообразие моделей процессов. Включение в эти модели процессов элементов организационной модели позволяет определить, какие компоненты имеют связь с другими

компонентами в рамках выполнения процесса. Диаграмма коммуникаций предоставляет возможность группировать все процессы в соответствии со взаимосвязями организационных единиц;

- **диаграммы классификаций**, позволяющие классифицировать функции, привязывая их к классам типов объектов. Классификация может проводиться по различным критериям;

- **диаграммы входа/выхода**, предоставляющие возможность описать входные и выходные данные, а также носители информации. Для данного типа модели в каждую ячейку может быть помещен только один графический символ, т. е. каждое поле отделяется от других полей сплошной линией. Верхняя строка содержит данные или носители информации, которые создаются функцией (являются ее выходом). Аналогично, в левой колонке находятся данные или носители информации, которые «входят» в функцию (являются ее входом);

- **диаграммы класса**. Расширенная объектно-ориентированная концепция ARIS позволяет связать функцию класса с информационными объектами. Это означает, что все информационные объекты, которые определяются в модели данных ARIS, могут получить символ класса, что достигается назначением им функций классов (методов) и описательных атрибутов (содержимое данных). Это описание выполняется с помощью диаграмм классов. В процессе привязки диаграммы класса к информационному объекту функция класса назначается данному информационному объекту. Диаграммы классов однозначно привязываются к отдельным информационным объектам и содержат следующие элементы (графические символы для типов объектов в архитектуре ARIS):

- информационный объект, описанный как класс;
- список атрибутов, привязанный к классу;
- список событий, произошедших согласно конкретному состоянию класса;
- список функций функциональной модели, которые привязаны к классу и переключаются событиями или сами переключают события;

- **матрицы выбора процессов**, отображающие различные сценарии выполнения процесса при помощи привязки основных процессов к отдельным сценариям. При моделировании матрицы выбора процесса используются следующие типы графических символов:

- сценарий;
- процесс;
- главный процесс;

- **диаграммы определения ответственности и авторизации доступа**. При объединении модели данных и организационной модели на уровне спецификации проекта должны быть решены следующие основные задачи:

- какие организационные единицы ответственны за конкретные объекты данных в компании;
- каким организационным единицам разрешен доступ к отдельным объектам данных;

- **диаграммы описания рабочего места**. Вопрос местонахождения (рабочего места) может быть разрешен посредством установки связей элементов организационной модели с типами прикладных систем, типами модулей и типами ИТ-функций;

- **структурная схема программы**. Схема позволяет моделировать все взаимосвязи типов прикладных систем, типов модулей и типов ИТ-функций при помощи других типов диаграмм ARIS — без характерного для ARIS разбиения на частные модели;

- **блок-схема программы**. Блок-схема программы описывает последовательность процедур в программе. Порядок процедур определяется связями (отношениями) между объектами. Эта блок-схема не представляет никаких данных;

- **диаграмма экрана**. Диаграмма экрана используется для описания экранных форм при разработке программного обеспечения с целью автоматизации процесса порождения экранных форм из диаграмм экранов.

Подробное описание вышеназванных диаграмм содержится в описании инструментариев ARIS. В методологии ARIS имеется поддержка объектного моделирования и методологии UML.

3.6. Модели ресурсов ARIS

Представление потоков материалов при моделировании бизнес-процесса (диаграммы eEPC и PCD с потоком материалов) осуществляется посредством связывания потоков с отдельными функциями бизнес-процесса в виде входа и выхода функций. Аналогично связи информационных объектов с функциями (преобразование информации представляется посредством функций) эта связь описывает преобразование типов материалов, поступающих на вход функции, в типы материалов на выходе функции. Кроме того, последовательности процессов дают возможность включить информацию по техническим ресурсам, которая необходима для преобразования материалов. В этом контексте будем различать операционные ресурсы, складское оборудование, транспортные системы и операционное обеспечение.

При помощи диаграммы типа «Технические ресурсы» эти ресурсы можно иерархически упорядочить, присвоить им тип и классифицировать. Существуют следующие типы объектов:

- **операционные ресурсы** — экземпляры различных типов операционных ресурсов, которые доступны для выполнения задач, стоящих перед компанией. Операционные ресурсы часто идентифицируются с помощью различных инвентарных номеров (например, номер завода);

- **тип операционных ресурсов**, представляющий совокупность различных видов операционных ресурсов, которые имеют одинаковую технологическую базу;

- **классы операционных ресурсов**, представляющие собой объединение типов операционных ресурсов. Их схожесть определяется в соответствии с различными критериями классификации. Следовательно, один тип операционных ресурсов может соответствовать нескольким классам операционных ресурсов;

- **складское оборудование** — экземпляры различных типов складского оборудования, доступного и предназначенного для выполнения стоящих перед компанией задач. Оборудование часто идентифицируется присвоенными ему номерами складов;

- **тип складского оборудования**, представляющий совокупность нескольких видов складского оборудования, которые имеют одинаковую технологическую базу;

- **классы складского оборудования**, представляющие собой объединение типов складского оборудования. Их схожесть определяется в соответствии с различными критериями. Следовательно, один тип складского оборудования может соответствовать нескольким классам складского оборудования;

- **техническое операционное обеспечение** — экземпляр типа технического операционного обеспечения. В общем случае он идентифицируется посредством инвентарного номера:

- **тип технического операционного обеспечения**, представляющий совокупность видов технического операционного обеспечения, которые имеют одинаковую технологическую базу;

- **классы технического операционного обеспечения**, представляющие собой объединение типов технического операционного обеспечения. Разделение на классы происходит в соответствии с различными критериями классификации. Следовательно, один тип технического операционного обеспечения может соответствовать нескольким его классам;

- **транспортная система** — экземпляр типа транспортной системы. В общем случае она может быть идентифицирована инвентарным номером или номером предприятия:

- **тип транспортной системы**, представляющий совокупность некоторых видов транспортных систем, которые имеют одинаковую технологическую базу;

- **классы транспортных систем** — объединение типов транспортных систем. Объединение в классы осуществляется в соответствии с различными критериями классификации. Следовательно, один тип транспортной системы может соответствовать нескольким классам транспортных систем.

Возможности иерархической организации диаграммы типа «Технические ресурсы» позволяют описывать структуру технически сложных объектов, 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». Данные типы приводились при описании диаграммы типа «Организационная схема». Эти типы объектов могут быть связаны с типами объектов «Операционные ресурсы», «Складское оборудование», «Техническое операционное обеспечение» и «Транспортная система».

На рис. 3.25 приведена диаграмма типа «Технические ресурсы».

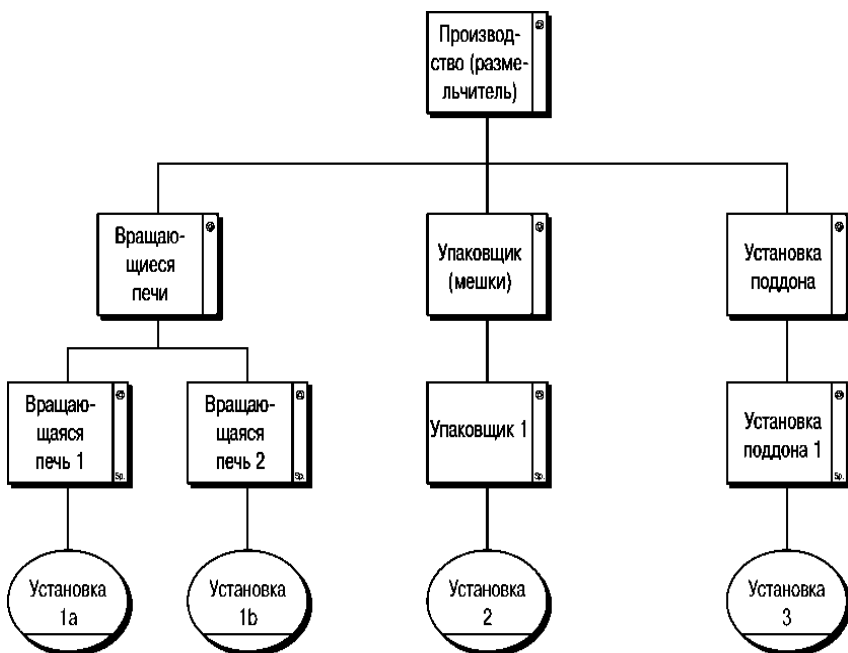


Рис. 3.25. Диаграмма типа «Технические ресурсы»

Моделирование потоков материалов. Диаграмма материалов. С помощью диаграммы материалов можно определять типы материалов (совокупность материалов, имеющих одинаковые характеристики), упорядочивать их в соответствии с иерархией и классифицировать по классам материалов. Типы материалов могут объединяться, образуя класс материалов. При таком объединении проблема схожести (отнесение материала к некоторому классу) решается с помощью различных критериев классификации. Другими словами, один тип материала может соответствовать нескольким классам материалов.

Типы материалов могут быть связаны с типами упаковочных материалов. Это означает, что определенные типы материалов могут перемещаться, только 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.

На рис. 3.26 приведена диаграмма материалов, включающая иерархию уровней и классификацию.

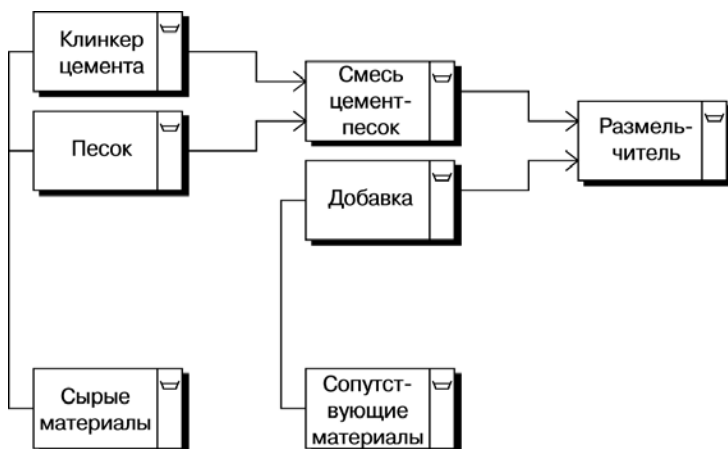


Рис. 3.26. Диаграмма материалов

Модель данных стоимостей, вычисленных по методу ABC¹. **Диаграмма стоимостных драйверов (СД).** Область применения диаграмм СД связана с определением стоимости по методу ABC. Диаграмма СД описывает иерархию стоимостных драйверов. **Стоимостный драйвер** — это имеющий определенный смысл элемент некоторой измеренной/референсной величины, предназначенный для оценки стоимости отдельных операций (функций) в рамках бизнес-процесса. Референсная величина представляет собой значение, которое получено из доступных источников информации и находится в постоянной зависимости (пропорции) от стоимостных оценок.

Стоимостные драйверы определяются только для процессов, производительность которых зависит от объемов или индуцированных процессов. Стоимостные драйверы не могут быть определены для таких процессов, как «Управление департаментом». Примером стоимостного драйвера может служить «Длина улицы» в процессе «Покрытие улицы асфальтом».

Иерархия стоимостных драйверов представляется в диаграмме СД с помощью направленных соединительных линий, имеющих тип «Определяет объем». Атрибуты «Числитель отношения СД» и «Знаменатель отношения СД» должны сопровождать эту связь. Если не определен «Знаменатель отношения СД», то его значение принимается за единицу. Частное от деления указанных атрибутов определяет количество связей между обоими стоимостными драйверами для вычисления стоимостных характеристик процесса.

Диаграмма СД представлена на рис. 3.27. В приведенном примере имеются два стоимостных драйвера: «Количество автомобилей (лимузинов)» и «Количество дверей». Чтобы показать, что каждый лимузин имеет 4 двери, атрибут «Числитель отношения СД» должен установить значение 4 на линии, соединяющей стоимостной драйвер «Количество автомобилей (лимузинов)» со стоимостным драйвером «Количество дверей».

¹ ABC (Activity Based Costing) — функционально-стоимостной анализ.

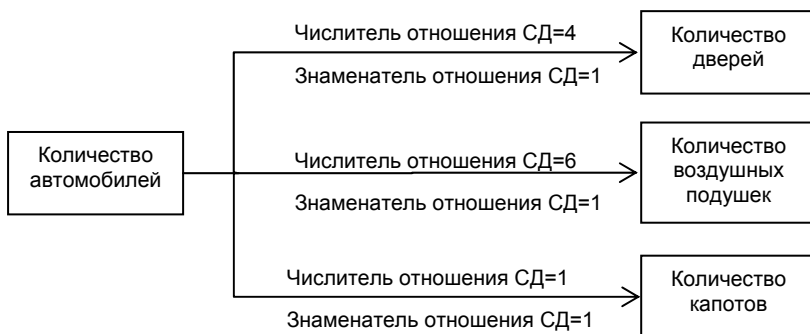


Рис. 3.27. Диаграмма стоимостных драйверов

Стоимостные драйверы связаны с конкретным процессом через управляющую модель. Множитель для каждой функции в рамках процесса определяется автоматически в соответствии с иерархией стоимостных драйверов.

Диаграмма стоимостных категорий. Иерархия стоимостных категорий представляется с помощью диаграмм стоимостных категорий. Стоимостные категории служат для систематического структурирования всей стоимостной информации, которая появляется в результате создания или оценки стоимостных драйверов (результатов).

Все стоимости могут быть структурированы в соответствии с различными критериями. Если стоимости разбиваются по производственным факторам, то это приводит к структурированию стоимости персонала (зарплата, комиссионные и т. п.), стоимости материалов (стоимость заготовок, стоимость электроэнергии) и издержек, связанных с налогами и другими выплатами.

Стоимостные категории могут также определяться в соответствии с наиболее важными операциями, такими как стоимость закупок, стоимость хранения на складе, стоимость производства, административные издержки, стоимости продаж. Обе описанные структуры могут быть еще более детализированы. Иерархия стоимостных категорий представляется направленными линиями, имеющими тип «Является вышестоящим».

Наиболее важным атрибутом для стоимостных категорий является шкала результата. Он описывает единицы, в которых измеряется результат стоимостной категории, например стоимость оплачиваемых часов или квадратных метров в комнате.

На рис. 3.28 показана диаграмма стоимостной категории производственных факторов с подструктурой, описывающей стоимость персонала. В ARIS ABC-диаграмма стоимостной категории может быть связана с экземпляром вычислений стоимости по методу ABC.

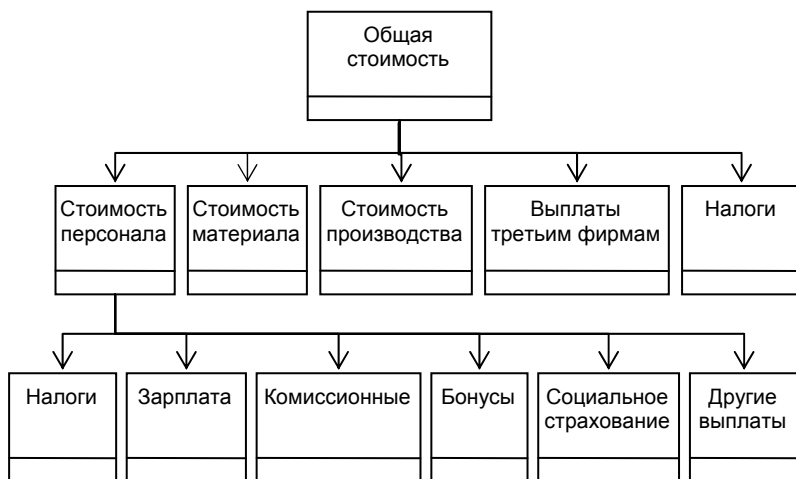


Рис. 3.28. Диаграмма стоимостной категории

Модель стоимостных центров перечисляет все стоимостные центры, которые будут проанализированы в экземпляре вычислений стоимости по методу ABC. Некоторые стоимостные категории и необходимый набор значений могут определяться для каждого стоимостного центра соответственно методу вычислений, тарифам, объемам.

3.7. Метод управления знаниями в методологии ARIS

Целью управления знаниями является систематическое обновление знаний как ресурса компании, важность которого постоянно возрастает. Термин **«управление знаниями»** охватывает разработку, мониторинг, поддержку и совершенствование стратегий, процессов, организационных структур и технологий для эффективной обработки знаний в рамках компании. Он включает также все действия, имеющие отношение к приобретению знаний, их представлению, передаче и использованию. Действия по управлению знаниями обычно не совершаются в изоляции. Они сопровождают, главным образом, оперативные и директивные бизнес-процессы в компании. Следовательно, на бизнес-процессы, обработку знаний, организационные структуры, информационные системы и другое необходим интегральный взгляд. Большинство из этих аспектов может быть отражено с помощью таких известных методов ARIS, как eEPCs, организационные схемы, диаграммы описания функций, eERMs и др.

Однако если цель заключается в точном представлении, анализе и совершенствовании управления знаниями, необходимы дополнительные средства представления для идентификации и структурирования содержимого соответствующих категорий знаний, описания и распределения знаний в пределах организации, моделирования создания и использования знаний в бизнес-процессах. По указанной причине добавляются два новых типа объектов: **«Категория знаний»** и **«Документированные знания»**, а также два новых типа моделей:

- 1) диаграмма структуры знаний;
- 2) карта знаний.

Кроме того, существующие типы моделей для представления бизнес-процессов расширяются конструкциями для управления созданием и использованием знаний.

Новые типы объектов и моделей полностью и методично интегрируются в наиболее важные типы моделей на уровне

формулировки требований (такие как eERM, «Организационные схемы» и «Функциональные деревья»), предоставляя возможность для дальнейшей интеграции. Это позволяет использовать модели оптимизации бизнес-процессов и применять их для анализа и совершенствования управления знаниями.

Диаграмма структуры знаний располагается в модели данных на уровне формулировки требований. Карта знаний, подобно расширенным типам моделей при моделировании бизнес-процессов, помещается в управляющую модель на уровне формулировки требований.

Тип объекта «Категория знаний», графически обозначаемый овалом с клапаном (рис. 3.29), представляет объект с содержимым, относящимся к конкретным знаниям. Примерами категорий знаний являются знание об управлении проектом, конкретные знания о производстве, конкретные знания о технологии, 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 знаний, могут быть:

- неявными знаниями, которые невозможно полностью документировать;
- знаниями о квалификации служащих или групп;
- явными знаниями, которые могут быть документированы в виде описаний или технической документации (чертежей).

Категории знаний часто содержат более одного вида знаний. Например, знания об управлении проектом включают опыт руководства проектом и советы по управлению проектом.

Знания могут быть описаны тремя основными атрибутами — «Описание», «Комментарий», «Источник» — и рядом дополнительных атрибутов, представленных в табл. 3.1.

В отличие от типа объекта «Категория знаний», охватывающего неявные и явные знания, тип объекта «Документированные знания» касается исключительно категории знаний, которая явно документирована или может быть в принципе документирована.

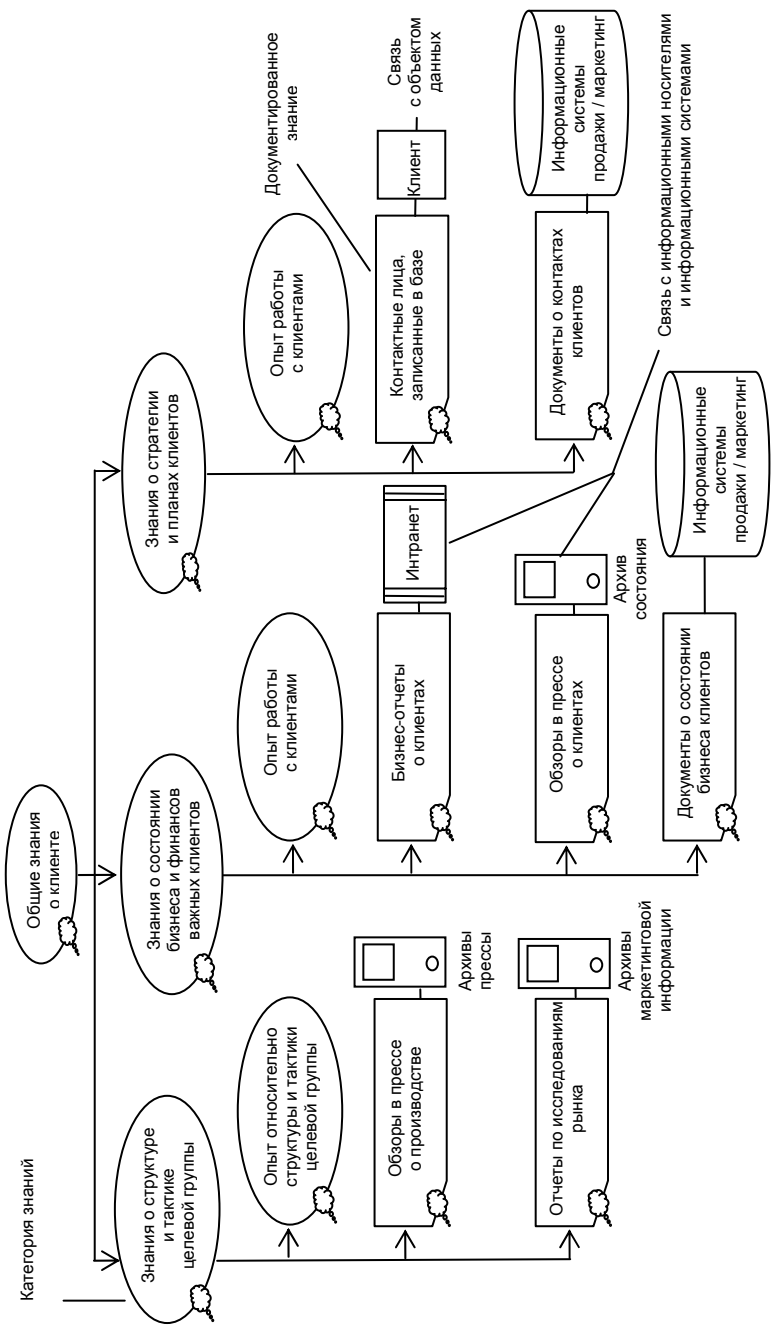


Рис. 3.29. Структурная диаграмма

Таблица 3.1

Описание дополнительных атрибутов знаний

Наименование атрибута	Элементы измерения	Описание / Пример
Частота корректировки	Градации: ежечасов, ежедневно, еженедельно, ежемесячно, ежегодно, редко, никогда	Частота корректировки описывает, как часто знания отдельной категории должны обновляться, чтобы отражать текущее состояние. Например, основные тригонометрические знания должны редко обновляться, если вообще должны, в то время как знания о ценах на товар необходимо обновлять ежедневно или ежедневно
Важность	Процент: 0–100	Важность может изменяться от 0 (совсем неважно) до 100 % (чрезвычайно важно)
Степень охвата	Процент: 0–100	Текущая степень охвата для отдельных знаний может меняться от 0 (совсем не охвачено) до 100 % (максимально возможный охват). Степень охвата категории знаний для отдельной организационной единицы или лица отображается в карте знаний с помощью соответствующего атрибута
Преимущество	Процент: 0–100	Относительное преимущество компании над конкурентами в терминах знаний может изменяться от 0 (конкуренты имеют максимально возможное преимущество над компанией) до 100 % (компания имеет максимально возможное преимущество над конкурентами)

Окончание табл. 3.1

Наименование атрибута	Элементы измерения	Описание / Пример
Использование знаний	Процент: 0–100	Степень применения отдельной категории знаний может изменяться от 0 (конкретные знания не используются совсем) до 100 % (оптимальное использование конкретных знаний)
Желаемая степень охвата	Процент: 0–100	Желаемая степень охвата отдельных знаний может изменяться от 0 (не охвачены совсем) до 100 % (максимально возможная степень охвата)
Важность в будущем	Градации: резкое уменьшение, уменьшение, постоянная, увеличение, резкое увеличение	Важность в будущем отображает ожидаемую тенденцию в изменении важности данной категории знаний для компании
Скорость структурных изменений	Процент: 0–100	Скорость структурных изменений отображает, насколько быстро должны изменяться методы, направленные на приобретение конкретных знаний (0 % не требуется изменений, 100 %; максимальная скорость структурных изменений)

Примером такого типа знаний является знание об использовании программного обеспечения, которое документировано в руководстве. При распределении информации по категориям осознание различий между основными категориями знаний и документированными знаниями способствует выявлению возможностей и ограничений в поддержке информационных систем, если только документированные знания могут быть сохранены, переданы или обработаны электронным способом.

Тип объекта «Документированные знания» обозначается прямоугольником с клапаном (см. рис. 3.29). Он содержит те же самые типы атрибутов, что и тип объекта «Категория знаний».

С помощью **структурной диаграммы знаний** (см. рис. 3.29) пользователь может поместить категории знаний в подгруппы, основываясь на их содержимом. Категория знаний может включать другие категории знаний, а также документированные знания. Документированные знания можно разделить на несколько подкатегорий документированных знаний, но они не могут включать общие знания.

Для документированных знаний структурная диаграмма знаний может отображать информационную среду, где знания документированы, или указывать, какие информационные объекты в модели данных (или классах объектно-ориентированных систем) служат для документирования этих знаний. Наконец, могут быть также смоделированы типы или классы прикладных систем для управления знаниями.

Распределение различных категорий знаний в рамках организации отображается с помощью **карты знаний**.

Типы объектов в организационной модели могут быть привязаны к категориям знаний с помощью соединения «Распорядиться». Кроме факта, что отдельный сотрудник или организационная единица обладает знаниями конкретной категории, может быть также описана и степень охвата. Соединение «Распорядиться» содержит атрибут «Степень охвата», который может иметь процентное выражение степени охвата знаний в категории знаний, выбранной для организационной единицы.

Карта знаний, приведенная на рис. 3.30, имеет ориентированную на организационные единицы структуру, т. е. соответствующая категория знаний связывается с каждой организационной единицей. Для карт знаний часто используется матричное представление. Это достигается организацией нескольких экземпляров одной и той же категории знаний в столбцы.

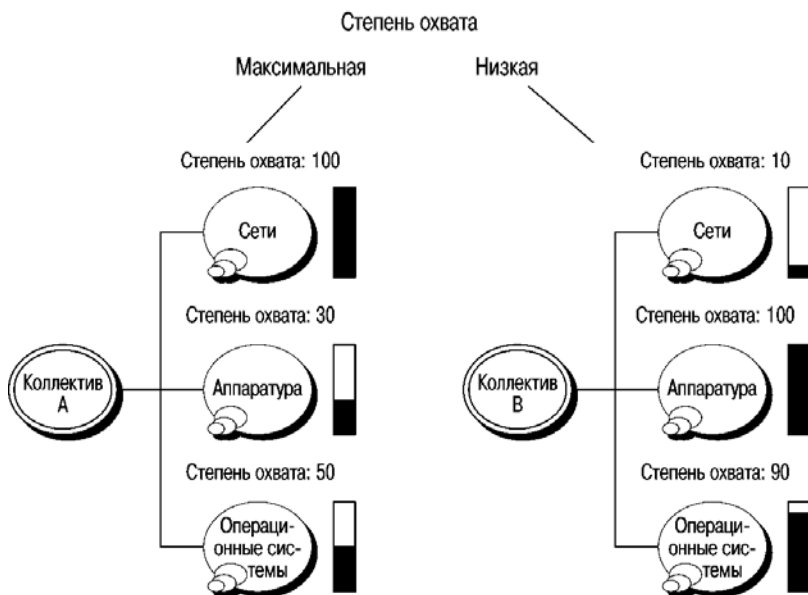


Рис. 3.30. Карта знаний, ориентированная на организационную единицу

При привязке категорий знаний к отдельным сотрудникам и оценке этих категорий необходимо учитывать, что сбор, документирование и, в особенности, электронная обработка данных, относящихся к служащим, — это предмет многих ограничений, связанных с законами и/или политикой компании. При создании, использовании или распространении карт знаний такого типа необходима уверенность в том, что это полностью согласуется с существующими законами или политикой компании.

Создание и применение знаний в бизнес-процессах компании моделируется с помощью типов моделей, используемых для представления бизнес-процессов (еEPC, еEPC с потоком материалов, «Офисный процесс», «Промышленный процесс», PCD, PCD с потоком материалов). Типы объектов «Категория знаний» и «Документированные знания» доступны в указанных диаграммах. Можно определить, какой тип знаний необходим для выполнения функции, и отметить, какие знания создаются и/или документируются, когда функция выполняется. Этот тип представления позволяет изучать бизнес-процессы в терминах обработки знаний. Например, может быть обнаружен пробел в необходимых знаниях и определен уровень повышения квалификации, требуемый при выполнении отдельной функции. Пример обработки знаний в модели еEPC показан на рис. 3.31.

3.8. Сравнительный анализ методологий ARIS и IDEF¹

В настоящее время на российском рынке представлено достаточно большое количество CASE-систем, многие из которых позволяют создавать описания (модели) бизнес-процессов предприятий. Очевидно, что выбор системы в значительной мере определяет весь дальнейший ход проекта. Рациональный выбор системы возможен при понимании руководством предприятия и его специалистами нескольких аспектов:

- целей проекта;
- требований к информации, характеризующей бизнес-процессы и необходимой для анализа и принятия решений в рамках конкретного проекта;
- возможностей CASE-систем по описанию процессов с учетом требований к разрабатываемой системе.

¹ При написании параграфа 3.8 использованы материалы В. Репина [26].

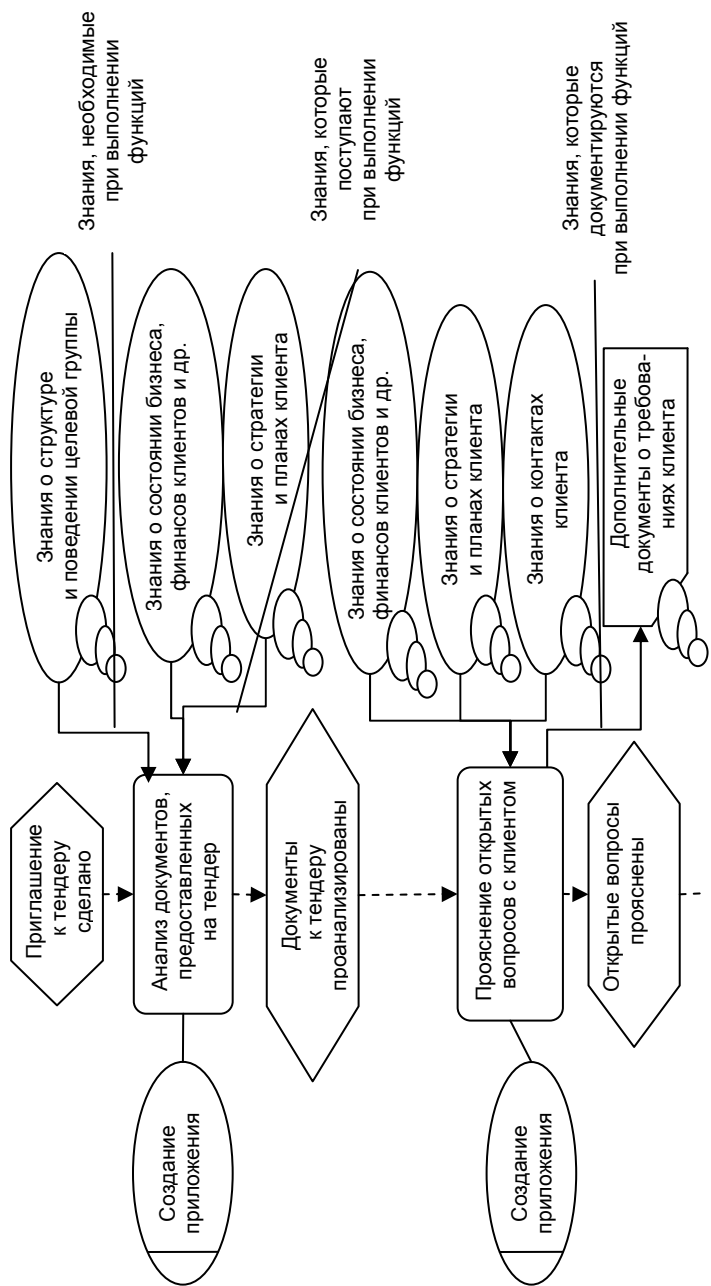


Рис. 3.3.1. Пример обработки знаний в модели eBRS

Говорить о преимуществе той или иной системы/нотации бессмысленно, пока не определены тип и рамки проекта, основные задачи, которые данный проект должен решить.

Описание бизнес-процессов проводится с целью их дальнейшего анализа и реорганизации. Целью реорганизации может быть внедрение информационной системы, сокращение затрат на выпуск продукции, повышение качества обслуживания клиентов, создание должностных и рабочих инструкций при внедрении стандартов ISO-9000 и т. д. Для каждой такой цели существуют параметры, определяющие набор критических знаний по бизнес-процессу. От задачи к задаче требования к описанию бизнес-процессов могут меняться. В общем случае модель бизнес-процесса должна давать ответы на следующие вопросы:

- какие процедуры (функции, работы) необходимо выполнить для получения заданного конечного результата;
- в какой последовательности выполняются эти процедуры;
- какие механизмы контроля и управления существуют в рамках рассматриваемого бизнес-процесса;
- кто выполняет процедуры процесса;
- какие входящие документы/информацию использует каждая процедура процесса;
- какие исходящие документы/информацию генерирует процедура процесса;
- какие ресурсы необходимы для выполнения каждой процедуры процесса;
- какая документация/условия регламентируют выполнение процедуры;
- какие параметры характеризуют выполнение процедур и процесса в целом.

Описание бизнес-процесса формируется при помощи нотации и инструментальной среды, которые позволяют отразить все указанные выше аспекты. Только в этом случае модель бизнес-процесса окажется полезной для предприятия, так как ее можно будет подвергнуть анализу и реорганизации.

Сравнительный анализ возможностей нотаций ARIS и IDEF приводится в табл. 3.2.

Одним из важнейших аспектов описания моделей бизнес-процессов является отражение на модели управляющих воздействий, обратных связей по контролю и управлению процедурой. В нотации ARIS eEPC управление процедурой может быть отражено только при помощи указания входящих документов, регламентирующих выполнение процедуры, и последовательности выполнения процедур во времени (запускающих событий). В отличие от ARIS, в нотации IDEF0 каждая процедура должна иметь хотя бы одно управляющее воздействие (вход управления — стрелка сверху). Если при создании модели в eEPC указывать только последовательность выполнения процедур, не заботясь об отражении управляющих воздействий (например, документов и информации), полученные модели будут иметь низкую ценность с точки зрения анализа и дальнейшего использования. К сожалению, именно эта ошибка наиболее распространена на практике. Создается модель WorkFlow (поток работы), отражающая простую последовательность выполнения процедур и входящих/исходящих документов, при этом управляющие (контрольные) воздействия на функции в модели не отражаются. Реальные процессы управления могут остаться «за кадром» на 30–90 % (рис. 3.32).

На рис. 3.32 функция 4 является контрольной и служит для проверки результатов выполнения работы, исполняемой функциями 2 и 3. Но данная модель не отвечает на следующие вопросы:

- каким образом осуществляется управляющее воздействие на функции 2 и 3, показан только тот факт, что по ходу процесса возможен возврат и повторное выполнение функций 2 и 3; информация об этой обратной связи может быть раскрыта только в виде описания в атрибутах объектов модели;
- какие документы (например, нормативы), распоряжения, внешние условия (например, влажность воздуха в помещении) регламентируют выполнение функций.

Таблица 3.2

Сравнительный анализ нотаций ARIS и IDEF

Критерии сравнения	ARIS	IDEF0	IDEF3
1. Принцип построения диаграммы / логика процесса	Временная последовательность выполнения процедур	Принцип доминирования	Временная последовательность выполнения процедур
2. Описание процедуры процесса	Объект на диаграмме	Объект на диаграмме	Объект на диаграмме
3. Входящий документ	Используется отдельный объект для описания («документ»)	Стрелка слева, стрелка сверху	Нет (может быть отражен в модели только привязкой объекта-комментария)
4. Входящая информация	Используется отдельный объект для описания («кластер», «технический термин»)	Стрелка слева, стрелка сверху	Нет (может быть отражен в модели только привязкой объекта-комментария)
5. Исходящий документ	Используется отдельный объект для описания («документ»)	Стрелка справа	Нет (может быть отражен в модели только привязкой объекта-комментария)
6. Исходящая информация	Используется отдельный объект для описания («кластер», «технический термин»)	Стрелка справа	Нет (может быть отражен в модели только привязкой объекта-комментария)

Окончание табл. 3.2

Критерии сравнения	ARIS	IDEF0	IDEF3
7. Исполнитель процедуры	Используется отдельный объект для описания («позиция», «организационная единица»)	Стрелка снизу	Нет (может быть отражен в модели только привязкой объекта-комментария)
8. Используемое оборудование	Используется отдельный объект для описания	Стрелка снизу	Нет (может быть отражен в модели только привязкой объекта-комментария)
9. Управление процедурой	Нет (может быть отражено только символами логики и событий (последовательность выполнения процедур) и/или указанием входящих документов)	Стрелка сверху	Только временная последовательность выполнения процедур и логика процесса
10. Контроль выполнения процедуры	Нет (может быть отражен указанием входящих документов)	Стрелка сверху	Нет
11. Обратная связь по управлению/контролю	Нет (может быть отражена только символами логики (последовательность выполнения процедур))	Стрелка сверху	Нет

Если пытаться отразить все условия и ограничения, определяющие выполнение функций, то потребуются описание большого количества событий и входящей информации (например, устных распоряжений руководителей) и модель станет сложной и плохо читаемой. Эти недостатки присущи также и нотации IDEF3. Указанных недостатков нет у нотации IDEF0. В то же время на моделях в IDEF0 не предусмотрено использование символов логики выполнения процесса. Таким образом, нотация ARIS eEPC является расширением достаточно простой нотации IDEF3. Для адекватного описания процесса управления в нотации eEPC необходимо заранее договориться, как будут отражены в модели документы (информация), регламентирующие выполнение процедур процесса.

Функциональные возможности инструментальных средств моделирования ARIS Toolset и AllFusion Process Modeler (ранее BPWin) можно корректно сравнивать только по отношению к определенному кругу задач.

В данном пособии рассматривается задача формирования моделей (описания) бизнес-процессов предприятия. Каждая из рассматриваемых систем имеет свои преимущества и недостатки. В зависимости от решаемых задач эти преимущества могут как усиливаться, так и наоборот. То же касается и недостатков: недостаток системы в рамках одного проекта может не быть недостатком в рамках другого.

Например, отсутствие четких соглашений по моделированию управляющих воздействий в рамках eEPC ARIS может привести к созданию моделей, не отвечающих на поставленные вопросы, в то время как нотация IDEF0 системы BPWin позволяет решить эту задачу. С другой стороны, описание процедуры, выполняемой одним сотрудником, может быть сделано более адекватно при помощи eEPC ARIS, чем IDEF0 или IDEF3 BPWin.

Сравнение функциональных возможностей систем приводится в табл. 3.3. Сравнивая две системы, следует сразу отметить, что для хранения моделей в ARIS используется объектная СУБД, и для каждого проекта создается новая база данных.

Таблица 3.3

Сравнение функциональных возможностей

Возможности/ инструментальная среда	ARIS Toolset 5.0	BPWin 4.0
1. Поддерживаемый стандарт	DFD (частично), ERM, UML	IDEF0, IDEF3, DFD
2. Система хранения данных модели	Объектная база данных	Модели хранятся в файлах
3. Ограничение на размер базы данных	Нет. Размер базы данных ограничивается вычислительными ресурсами	Нет. Размер базы данных ограничивается вычислительными ресурсами
4. Возможность групповой работы	Есть. Используется ARIS Server	Есть. Используется ModelMart
5. Ограничение на количество объектов на диаграмме	Нет	От 2 до 8
6. Возможность декомпозиции	Неограниченная декомпозиция. Возможна декомпозиция на различные типы моделей	Неограниченная декомпозиция. Возможен однократный переход на другую нотацию в процессе декомпозиции
7. Формат представления моделей	Не регламентируется	Стандартный бланк IDEF с возможностью его отключения
8. Удобство работы по созданию моделей	Сложная панель управления, есть выравнивание объектов, есть откат назад	Простая панель управления, нет выравнивания объектов, нет отката назад
9. UDP-свойства объектов, определяемые пользователем	Большое, но ограниченное количество свойств, количество типов ограничено	Количество UDP не ограничено. Количество типов ограничено

Окончание табл. 3.3

Возможности/ инструментальная среда	ARIS Toolset 5.0	BPWin 4.0
10. Возможность анализа стоимости процессов	Есть. Возможность использовать ARIS ABC	Упрощенный анализ стоимости по частоте использования в процессе. Возможность экспорта в Easy ABC
11. Генерация отчетов	Создание отчетов на основе стандартных и настраиваемых пользователем макросов Visual Basic	RPT Win, возможность визуальной настройки отчетов, включая расчет по формулам с использованием UDP
12. Сложность разработки нестандартных отчетов	Сложно	Просто

Для удобства пользователя модели объекты моделей могут храниться в различных группах, организованных в зависимости от специфики проекта. Вполне естественно, что в ARIS предусмотрены различные функции по администрированию БД: управление доступом, консолидация и т. п. В BPWin данные модели хранятся в файле, что существенно упрощает работу по созданию модели, но, с другой стороны, ограничивает возможности по анализу объектов модели. В ModelMart также предусмотрено администрирование базы данных.

Часто одним из недостатков BPWin сторонники методологии ARIS называют ограничение по количеству объектов на диаграмме. Однако опыт реальных проектов показывает, что для проекта, результаты которого можно реально использовать (критерий — обозримость), количество объектов в БД ARIS или модели BPWin составляет 150–300. Это означает, что при 8 объектах на одной диаграмме общее количество диаграмм (листов) в модели составит 20–40.

Базы данных ARIS Toolset (как и BPWin), содержащие более 500 объектов, фактически невозможно использовать. Следует подчеркнуть, что модель создается для выделения и анализа проблем, т. е. требуется детальное описание наиболее сложных, проблемных областей деятельности, а не тотальное описание всех процессов. Как ни странно, у директоров компаний существует вера в то, что детальное описание процессов само по себе представляет ценность и может решить многие проблемы. Это далеко не так. Именно понимание того, что нужно описывать и какие аспекты функционирования реальной системы при этом отражать, определяет успех проекта по моделированию бизнес-процессов.

ARIS предоставляет существенно больше возможностей по работе с отдельными объектами модели, но именно вследствие чрезмерного количества настроек работа по созданию модели должна регламентироваться сложной, многоаспектной документацией — соглашениями по моделированию. Разработка этих соглашений является сложной, дорогостоящей задачей, требующей значительного времени (1–3 месяца) и наличия квалифицированных специалистов. Если проект с использованием ARIS начинается без детальной проработки таких соглашений, то вероятность создания моделей бизнес-процессов, не отвечающих на поставленные вопросы, составляет 80–90 %.

BPWin отличается от ARIS простотой в использовании и достаточно строгой регламентацией при создании диаграмм (наличие стандарта IDEF и рекомендаций по его применению, бланка IDEF для создания диаграммы, ограничений по количеству обязательно заполняемых полей и объектов на одной диаграмме и др.). ARIS, безусловно, является более «тяжелым» инструментом по сравнению с BPWin, но это в итоге оборачивается значительными трудностями и высокими затратами на его эксплуатацию.

Позиционирование систем BPWin и ARIS можно провести по функциональным возможностям и простоте использования в проекте (рис. 3.33).

Для ведения небольших по масштабам (малые и средние предприятия, 2–5 человек в группе консультантов) и длительности (2–3 месяца) проектов рационально использовать BPWin.

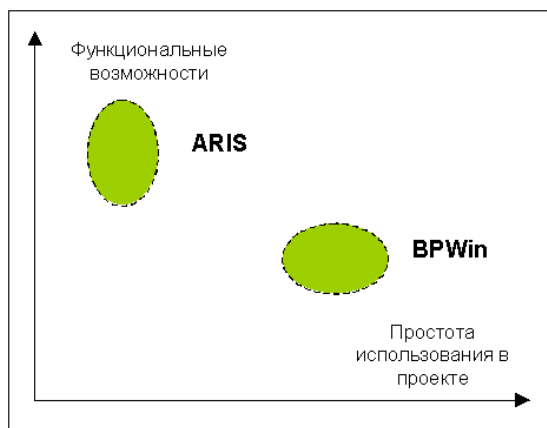


Рис. 3.33. Позиционирование систем BPWin и ARIS по функциональным возможностям и простоте использования в проекте

Для крупных и/или длительных проектов (например, внедрение систем непрерывного улучшения бизнес-процессов, ISO, TQM) больше подходит ARIS. В этом случае подготовительные работы по созданию регламентирующей документации могут занять 1–3 месяца, но это является необходимым элементом последующей успешной работы.

Компанией IDS Scheer в 2009 г. создан продукт ARIS Express, ориентированный на новичков в моделировании процессов и обычных пользователей, не достаточно глубоко разбирающихся в информационных технологиях, а также на преподавателей и студентов университетов. Программное обеспечение ARIS Express предлагается в качестве альтернативы таким инструментам для рисования, как MS Visio и MS PowerPoint. ARIS Express доступен для загрузки на сайте сообщества ARIS Community. В настоящее время продукт существует только с англоязычным интерфейсом.

Вопросы для самоконтроля

1. Кто автор концепции ARIS, и какие преимущества этой концепции он декларирует?
2. Укажите четыре отдельных типа, отражающие различные аспекты исследуемой системы, на которые делится модель ARIS.
3. С какой целью в ARIS вводится управляющая модель?
4. Как строятся модели архитектуры ARIS при описании бизнес-процесса?
5. Опишите общую архитектуру ARIS.
6. С какой целью вводится организационная модель ARIS?
7. Опишите организацию календаря смен в ARIS.
8. Опишите процессно-ориентированное функциональное дерево в ARIS.
9. Опишите Y-диаграмму в ARIS.
10. Приведите пример диаграммы целей в ARIS.
11. С какой целью в ARIS вводится функциональная модель?
12. Опишите расширенную модель «сущность-отношение» в ARIS.
13. Приведите пример описания объектов на уровне формулировки требований и спецификации проекта в информационной модели ARIS.
14. Приведите описание и пример диаграмм PCDs.
15. Приведите описание и пример диаграмм EPCs.
16. Какие диаграммы могут использоваться в ARIS при построении управляющей модели (за исключением PCDs и EPCs)? Дайте им краткую характеристику.
17. Приведите описание и пример диаграмм «Технические ресурсы».
18. Приведите описание и пример диаграмм материалов.
19. Приведите описание и пример диаграмм стоимостных драйверов.
20. Что собой представляет метод управления знаниями в методологии ARIS?
21. Приведите сравнительный анализ методологий ARIS и BPwin.

ОБЗОР МОДЕЛЕЙ И МЕТОДИК ПОСТРОЕНИЯ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ

4.1. Модель Захмана

Дж. Захман (John A. Zachman) сделал наиболее значимый вклад в развитие концепции архитектуры предприятия. Модель Захмана (Zachman Framework for Enterprise Architecture) постоянно эволюционирует и является основой, на базе которой многие организации создают свои собственные методики описания информационной инфраструктуры предприятия. Модель Захмана послужила основой для создания целого ряда методик и моделей описания архитектуры предприятия:

- Федеральной архитектуры США (FEAF, Federal Enterprise Architecture Framework);
- методики описания архитектуры Open Group (TOGAF, The Open Group Architecture Framework);
- методики описания архитектуры Министерства обороны США (DoDAF, Department of Defence Architecture Framework) и др.

Первая версия модели Захмана была опубликована в 1987 г. и предназначалась большей частью для ИТ-систем. Эта версия модели в течение 1992–1996 гг. была расширена и обобщена в разрезе описания архитектур сложных производственных систем любого типа, найдя применение во многих компаниях, входящих в список 2000 крупнейших корпораций мира (General Motors, Bank of America и др.) [1].

Модель Захмана⁵ основана на классических методологиях построения архитектуры предприятия. Модель обеспечивает общий словарь и набор перспектив, или структур, для описания современных сложных корпоративных систем. Дж. Захман определяет *архитектуру предприятия* как набор описательных представлений (моделей), применимых для описания предприятия в соответствии с требованиями управленческого персонала и способных развиваться в течение определенного периода. Термин «архитектура» здесь не случаен, он подчеркивает существующую аналогию между внутренней структурой абстрактного объекта — предприятия — и сложным искусственным объектом, таким как здание или орбитальная международная космическая станция (МКС) [1].

Модель Захмана преследует две основные цели:

- 1) логически разбить все описание архитектуры на отдельные разделы для упрощения их формирования и восприятия;
- 2) обеспечить возможность рассмотрения целостной архитектуры с выделенных точек зрения или соответствующих уровней абстракции.

В период опубликования работ Захмана в качестве традиционного подхода при формировании описания системы использовалась концепция «жизненного цикла», включающего такие этапы, как *планирование, анализ, проектирование, разработка, документирование, внедрение и промышленная эксплуатация*. На каждом из этих этапов рассматриваются вопросы, связанные как с функциями системы, так и с данными. Захман предложил вместо традиционного подхода, связанного с рассмотрением отдельных аспектов работы системы в различные моменты времени, использовать рассмотрение системы с различных перспектив.

Основная идея заключается в том, чтобы обеспечить возможность последовательного описания каждого отдельного аспекта системы в координации со всеми остальными. Собственно модель представляется в виде таблицы (табл. 4.1).

⁵ По глубине подхода и значимости модели, скорее, должен был быть применен перевод оригинала «framework» как «методики» [1].

Таблица 4.1

Модель Захмана

	Данные ЧТО?	Функции КАК?	Сеть ГДЕ?	Организации КТО?	Расписание КОГДА?	Стратегии ПОЧЕМУ?	
Планировщик (1-й уровень)	Список важных понятий и объектов	Список основных бизнес-процессов	Список мест нахождения	Список организаций, важных для бизнеса	Список важных событий	Список бизнес-целей и стратегий	Сфера действия (контекст)
Владелец, менеджер (2-й уровень)	Концептуальная модель данных	Модель бизнес-процессов	Схема логистики	Модель потока работ (workflow)	Календарный план реализации	Бизнес-план	Концептуальная модель предприятия
Конструктор, архитектор (3-й уровень)	Логические модели данных	Архитектура приложений	Модель распределенной архитектуры	Архитектура интерфейса пользователя	Структура процессов	Конкретизация ролей и бизнес-правил	Системная (логическая) модель
Проектировщик (4-й уровень)	Физическая модель данных	Системный проект	Технологическая архитектура	Архитектура презентации	Структуры управления	Реализация ролей и бизнес-правил	Технологическая (физическая) модель
Разработчик (5-й уровень)	Описание структуры данных	Программный код	Сетевая архитектура	Архитектура безопасности	Определение временных привязок	Реализация бизнес-логики	Детали реализации
Пользователь (6-й уровень)	Фактические базы данных	Исполняемый код и инструкции к функциям	Описание взаимодействия в сети	Обученный персонал	Список фактических бизнес-событий	Работающие правила	Оценка функционирования

Перспективы (строки в таблице) соответствуют различным уровням управления предприятием, если речь идет об архитектуре предприятия или использовании ИС:

- **первый уровень** соответствует уровню интересов высшего руководства и собрания акционеров. В применении к деятельности предприятия — это **верхняя строка таблицы, представляющая, по сути, контекст модели**. В данной строке демонстрируется планирование бизнеса в целом (бизнес-модель). На этом уровне вводятся достаточно общие основные понятия, определяющие бизнес (например, продукты и услуги, клиенты, расположение объектов бизнеса), а также формулируется бизнес-стратегия (колонка «Стратегия»). Данная строка определяет контекст всех последующих строк;

- **второй уровень** соответствует интересам бизнес-менеджеров и владельцев процессов, на нем определяется концептуальная модель, которая предназначена для описания в терминах бизнеса структуры организации, ключевых и вспомогательных бизнес-процессов. Две верхние строки соответствуют наиболее общим представлениям и достаточно широко описывают существующее окружение, планы и цели;

- **третий уровень** — уровень, на котором происходит организация «командной» работы бизнес-менеджеров, бизнес-аналитиков и менеджеров, отвечающих за разработку ИТ. Это уровень логической модели, здесь бизнес-процессы описываются уже в терминах информационных систем, включая различные типы данных, правила их преобразования и обработки для выполнения определенных на втором уровне бизнес-функций;

- **четвертый уровень и последующие** описывают детали, представляющие интерес для ИТ-менеджеров, проектировщиков, разработчиков. На нем определяется технологическая модель, включающая физическую модель и детали реализации, т. е. осуществляется привязка данных и операций над ними к выбранным технологиям реализации. Например, здесь может быть определен выбор реляционной СУБД, средств работы с неструктурированными данными и/или объектно-ориентированной среды;

- **пятый уровень** соответствует детальной реализации системы, включая конкретные модели оборудования, топологию сети, производителя и версию СУБД, средства разработки и собственно готовый программный код. Многие из работ на данном уровне часто выполняются субподрядчиками;

- **шестой уровень** описывает работающую систему. На этом уровне могут быть введены такие объекты, как инструкции для работы с системой, фактические базы данных.

На каждом уровне участники рассматривают одни и те же **категории вопросов**, соответствующие столбцам (колонкам) таблицы, только с различным уровнем абстракции и детализации.

Колонка «Данные» (ответ на вопрос «ЧТО?») определяет используемые в системе данные. *На верхнем уровне* достаточным будет простое перечисление основных объектов, используемых в бизнесе. *На втором уровне* данные (объекты) объединяются в семантическую модель высокого уровня и обычно описываются в виде диаграммы «сущности-связи» с отражением основных связей и наиболее существенных бизнес-ограничений. *На третьем уровне* эта модель приводится к нормализованной форме, определяются все атрибуты и ключи. *Четвертый уровень* представляет собой физическую модель данных в системе (в объектно-ориентированном подходе — иерархию классов). *Пятый уровень* содержит описание модели на языке управления данными для формирования таблиц, готовые библиотеки классов, табличные пространства СУБД. *Шестой уровень* может описывать фактические наборы данных, в том числе такие характеристики, как журналы доступа, размеры реально занимаемого дискового пространства, статистика обращений и т. п. Можно отметить определенное несовершенство данной модели при использовании объектно-ориентированного подхода — фактически модель предписывает раздельное рассмотрение данных (свойств) и функций (методов) классов.

Колонка «Функции» (ответ на вопрос «КАК?») предназначена для описания последовательной детализации способов реализации миссии предприятия на уровне отдельных операций. *На первом уровне* достаточным будет простое перечисление

бизнес-процессов. *Второй уровень* будет содержать модель бизнес-процессов, которая впоследствии детализируется *на третьем уровне* в операции над данными и архитектуру приложений; *на четвертом уровне* — в методы классов; *на пятом уровне* содержится программный код и, наконец, исполняемые модули — *на шестом уровне*. При этом, начиная с четвертого уровня, рассмотрение ведется уже не в рамках предприятия в целом, а по отдельным подсистемам или приложениям.

Колонка «Сеть» (ответ на вопрос «ГДЕ?») определяет пространственное распределение компонентов системы и сетевую организацию. *На уровне планирования бизнеса* достаточно определить расположение всех производственных объектов. *На втором уровне* эти объекты объединяются в модель со связями, характеризующими взаимодействие между собой, — будь то обмен информацией или поставки товаров. *На третьем уровне* системной архитектуры осуществляется привязка компонентов информационной системы к узлам сети. *Четвертый уровень* служит для определения физической реализации в терминах аппаратных платформ и системного программного обеспечения, используемых для интеграции различных компонентов ИС между собой. Типичным примером могут являться брокеры запросов или средства обмена сообщениями. *На пятом уровне* определяются используемые протоколы и спецификации каналов связи. *На шестом уровне* описывается функционирование реализованной сети.

Колонка «Организации» (ответ на вопрос «КТО?») определяет участников процесса. *На уровне планирования бизнеса* здесь представлен список подразделений предприятия и выполняемые ими функции. *На втором уровне* приводится полная организационная диаграмма, а также могут быть определены общие требования к информационной безопасности. Далее последовательно определяются участники бизнес-процессов и их роли (*уровень 3*), требования к интерфейсам пользователя и правила доступа к отдельным объектам (*уровень 4*), их физическая реализация на уровне кода или операторов определения доступа к таблицам в СУБД (*уровень 5*). *Шестой уровень* описывает обученных пользователей системы.

Колонка «Расписание» (ответ на вопрос «КОГДА?») определяет временные характеристики бизнес-процессов и работы системы. Детализация осуществляется сверху вниз, начиная от списка важных событий (*уровень 1*) и календарного плана (*уровень 2*), характеризующих выполнение процессов (например, требование ко времени оформления сделки). *На третьем уровне* определяются события, вызывающие изменение состояния информационных объектов и инициацию операций над ними (диаграммы зависимостей, последовательностей). *На четвертом уровне* эти события транслируются в программные вызовы (триггеры) или передаваемые сообщения (диаграмма потоков управления). *Пятый уровень* определяет физическую реализацию обработки таких событий (определения интервалов, временные диаграммы), *шестой уровень* представляет историю функционирования системы.

Колонка «Стратегии» (ответ на вопрос «ПОЧЕМУ?») служит для определения мотивации и задает порядок перехода от задач бизнеса к требованиям и элементам ИС. Исходной точкой является бизнес-стратегия (*уровень 1*), транслируемая последовательно в бизнес-план (*уровень 2*), правила и ограничения для реализации бизнес-процессов (*уровень 3*), соответствующие приложения, необходимые для включения в состав ИС и в дальнейшем в их физическую реализацию (*четвертый уровень*).

Таблица заполняется по следующим правилам:

- каждая клетка таблицы независима от других, вместе они образуют функционально полное пространство для описания системы («базис»);
- каждая клетка содержит соответствующее описание аспекта реализации системы в виде определенной модели или, возможно, простого описания (текстового документа);
- порядок следования колонок несущественен;
- базовые модели для каждой колонки уникальны;
- модели в клетках каждого ряда в совокупности образуют полное описание системы с выбранной перспективы;
- заполнение клеток должно проводиться последовательно «сверху вниз».

Модель Захмана является одной из наиболее продвинутых сред в части гармоничного и комплексного учета всех архитектурно-существенных факторов, позволяя при этом концентрироваться на отдельных аспектах архитектуры и сохраняя общий взгляд на предприятие как единое целое. Модель легка для понимания, логически полна и согласованна, нейтральна по отношению к инструментарию, что обуславливает ее широкое применение. Модель Захмана не поддерживает представление о динамике развития организации и ее ИС (отсутствие оси времени), является достаточно поверхностной в смысле степени детализации референсной моделью, достаточно бедна с технических позиций [4].

Существуют специализированные продукты, например *Popkin Software Architect*, которые фактически основаны на модели Захмана и позволяют достаточно эффективно управлять созданием моделей и артефактов описания архитектуры предприятия.

Обобщение подхода Захмана было предложено в работах Е.З. Зундера [27]. Основная идея заключается в обеспечении возможностей отражения постоянного развития предприятия и его информационных систем как непрерывной последовательности трансформаций. Вместо традиционной двумерной таблицы было предложено ввести трехмерную схему, добавив к плоским схемам ось стратегического времени, на которой располагаются отрезки времени осуществления различных проектов и стадий развития информационных систем и всего предприятия.

Таким образом, была создана «объемная» схема архитектуры предприятия (модель «3D-предприятие»), которая строится в трех измерениях с учетом временного пространства. При этом первые два измерения аналогичны используемым Захманом, но не совпадают с оригиналом по содержанию и трактовке. Третья ось позволяет явно определять изменения, которые происходили и будут происходить с предприятием, его существующими информационными системами, а также с различными проектами развития и трансформации.

Следует отметить, что предложенный вариант развития исходного подхода Захмана не является единственно возможным.

Существует большое количество модельных схем, которые в той или иной мере используют данный подход, хотя визуальное представление модели в целом может достаточно сильно отличаться. Примерами таких схем являются модели *Extended Enterprise Architecture Framework (E2AF)* [28] и *4-доменной архитектуры (Four Domains architecture, FDA)* [29].

4.2. Модель описания ИТ-архитектуры Gartner

Одним из возможных достаточно простых форматов описания архитектуры Gartner является простое матричное представление, которое для каждой из основных областей архитектуры ИТ (данных, приложений, интеграции, общих сервисов и инфраструктуры) «последовательно накладывает» несколько спецификаций (бизнес-потребности, принципы, процессы и руководства, протоколы и стандарты, используемые продукты и технологии), отличающихся по уровню детализации и конкретизации. Данный подход позволяет обеспечить отслеживание логической связи между выбранными технологиями, их ценностью и потребностями для бизнеса.

В 2002 г. в компании Gartner была сформулирована новая концепция архитектуры предприятия, ставшая определенным обобщением ранней модели ИТ-архитектуры на основе матричного представления и косвенным отражением растущей важности вопросов взаимодействия предприятий. На формирование новой концепции также оказали влияние концепции сервис-ориентированной архитектуры и осознание факта существования различных стилей архитектуры информационных систем, соответствующих различным стилям бизнес-процессов.

Модель Gartner 2002 года сформулирована в виде четырех уровней (рис. 4.1):

1) *среда бизнес-взаимодействия (Business Relationship Grid)*, описывающая новую модель «виртуального» бизнеса и процессы,

связанные с кооперацией предприятий и бизнесом, B2B¹. Этот уровень идентичен понятию «отраслевой нервной системы» взаимодействующих предприятий. Он получил развитие в связи с распространением Интернета как среды взаимодействия и связан с понятиями доступа и межорганизационного взаимодействия;

2) **бизнес-процессы и стили бизнес-процессов** описывающие, каким образом организация выполняет свои ключевые функции (бизнес-процессы предприятия, такие как обработка заказа, мониторинг производственных процессов, анализ использования критически важных ресурсов, совместная работа с информацией);

3) **шаблоны**, описывающие модели и алгоритмы, которые могут широко использоваться для решения различных задач на предприятии. Отметим, что шаблоны охватывают не только область программного обеспечения, но и соответствующие сетевые и вычислительные ресурсы. Примерами шаблонов являются: трехуровневая архитектура прикладных систем (интерфейс — логика — данные), использование «толстого» клиента в архитектуре клиент/сервер, хранилища данных. Что касается приложений, то упор сделан на использовании **шаблонов сервис-ориентированной архитектуры**, т. е. реализации приложений в виде модульного набора различных типов сервисов. Это позволяет в перспективе интегрировать приложения как web-сервисы;

4) **технологические строительные блоки (кирпичики — bricks)**, соответствующие технологической архитектуре и включающие в себя операционные системы, серверы, базы данных, собственно данные и пр.

В данной схеме первый и второй уровни ориентированы на совместную работу бизнес-руководителей и ИТ-специалистов (бизнес-архитектура), а третий и четвертый уровни входят во внутреннюю компетенцию ИТ-службы. При этом уровни ИТ-архитектуры соответствуют различным уровням выполнения операций реального бизнеса (рис. 4.2).

¹ Под понятием B2B подразумеваются системы электронной коммерции (электронной торговли) — программно-аппаратные комплексы, являющиеся инструментами для осуществления торгово-закупочной деятельности в сети Интернет.

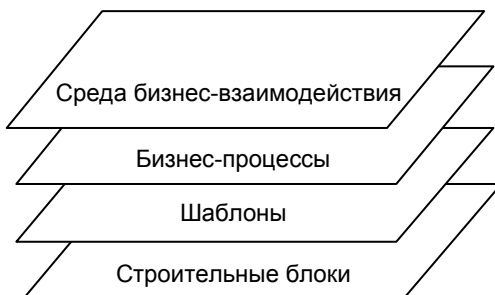


Рис. 4.1. Уровни модели архитектуры Gartner

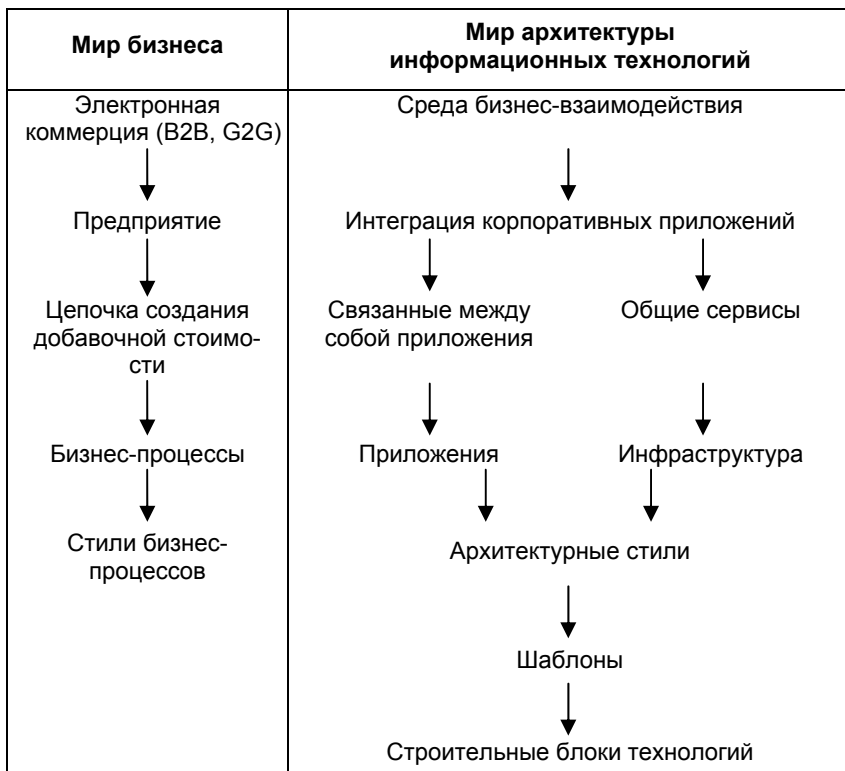


Рис. 4.2. Архитектура ИТ в бизнес-контексте

Такой подход позволяет продемонстрировать руководству предприятия механизм влияния решений в области ведения бизнеса на решения в области использования ИТ. Первый и второй уровни модели показывают важность и необходимость построения архитектуры предприятия по мере развития моделей ведения бизнеса в сторону «более виртуальных» структур («расширенных организаций»), успех которых будет в существенной степени зависеть от рациональной реализации архитектуры.

Полная модель Gartner представляет собой «трехмерную» взаимосвязанную комбинацию (треугольную пирамиду) бизнес-архитектуры, технологической и информационной архитектур.

Подход компании Gartner представляет собой пример реализации методологии достаточно высокого уровня. Он задает только общую рамочную модель описания и не определяет ни форматов, ни какого-либо специализированного языка для описания. Что касается разработки архитектуры, то в данном подходе сформулированы важные и полезные рекомендации в виде последовательности шагов и задач участников, которые не детализируются до уровня моделей процесса разработки архитектуры [1].

4.3. Методика META Group

Основу методики META Group составляет **Технологическая архитектура масштаба предприятия** (EWTA — Enterprise Wide Technical Architecture). Однако по мере осознания более тесной связи между бизнесом и информационными технологиями в представления архитектуры предприятия (домены или предметные области) в соответствии с методикой META Group были добавлены такие, как **бизнес-архитектура** (EBA — Enterprise Business Architecture), **архитектура информации** (EAI — Enterprise Information Architecture) и **портфель прикладных систем предприятия** (EAP — Enterprise Application Portfolio). Это соответствует эволюции понятия «Архитектура предприятия», которая происходила на рынке в целом и продолжается в принятой сегодня практике выделения доменов архитектуры.

Кроме того, расширяя другие модели и методики, методика META Group рассматривает архитектуру предприятия в интеграции с ключевыми процессами и проектами, такими как процесс управления корпоративными ИТ-программами, процесс выработки стратегии и планирования, проект EPM (Enterprise Program Management).

Объединяющим для всех доменов методики META Group является процесс формулировки бизнес-требований к ИТ-архитектуре, что оформляется в виде двух документов:

- 1) «Видение общих требований» (CRV — Common requirements Vision);
- 2) «Принципы концептуальной архитектуры» (CA — Conceptual Architecture).

Организация рабочего процесса разработки архитектуры и быстрое создание начальной версии архитектуры предприятия в соответствии с методикой META Group состоит в прохождении нескольких этапов:

- 1) разработка общих требований;
- 2) разработка концептуальной архитектуры;
- 3) разработка плана реализации.

Этап 1 — разработка общих требований — включает:

- анализ развития внешней среды (технологические тенденции);
- выбор бизнес-стратегий и основных движущих сил;
- формирование требований к информационным системам со стороны бизнеса;
- формирование требований к технологической архитектуре, обеспечивающей адекватные возможности для информационных систем с точки зрения потребностей бизнеса.

Документ, представляющий видение общих требований, может быть оформлен в виде таблицы (табл. 4.2).

Важным аспектом является документирование явных связей между бизнес-стратегией (потребностями бизнеса) и требованиями к информационным системам и в конечном итоге установление логических связей с требованиями к технологической архитектуре.

Таблица 4.2

Пример оформления общих требований

Тенденция	Формулировка требования		
	Бизнес-стратегия предприятия	Требования к ИС	Требования к архитектуре
Наличие задержки в предоставлении услуги для 20 % клиентов	Увеличение доли рынка, достигаемое за счет рациональной организации процесса обслуживания, позволяющей сократить время ожидания для клиента	Оперативная (незамедлительная) передача в производство информации о заказах, независимо от канала и места их получения	Наличие ИТ-инфраструктуры, обеспечивающей управляемый доступ и своевременную передачу информации с целью достижения операционной эффективности

Для установления логических связей рекомендуется использовать простые матрицы, представленные на рис. 4.3 [1]. Документированные связи послужат основой для принятия решений об инвестициях.

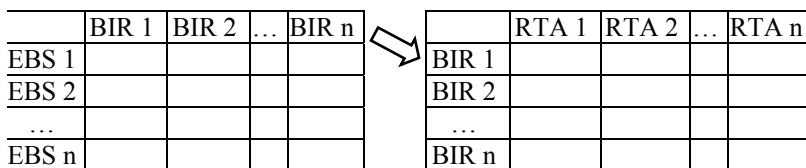


Рис. 4.3. Матрица связей между элементами архитектуры предприятия: бизнес-стратегиями (EBS — Enterprise Business Strategy); требованиями к ИС (BIR — Business Information Requirements); требованиями к технологической архитектуре (RTA — Requirements for Technical Architecture)

Видение общих требований агрегирует все требования к технологической архитектуре, и это служит основой для формулировки принципов концептуальной архитектуры.

Этап 2 — разработка концептуальной архитектуры, определяющей логически связанный набор принципов, использование которых обеспечивает общее руководство для развития ИС предприятия и технологической инфраструктуры. На этом этапе параллельно ведется и разработка наиболее приоритетных доменов архитектуры. Здесь же выполняется анализ на несоответствие между текущим и желаемым состояниями архитектуры.

Концептуальная архитектура разрабатывается еще до создания других архитектурных доменов и основана на принципах, суть которых состоит в следующем:

- принципы представляют собой утверждения, которые касаются архитектурного процесса или содержания архитектуры;
- количество принципов ограничено, и они являются неизменным фундаментом для построения архитектуры;
- принципы должны быть утверждениями, чья справедливость для организации имеет «вечный» характер, поскольку они задают систему ценностей для архитектуры в целом.

В соответствии с методикой META Group результатом разработки принципов концептуальной архитектуры является выделение в технологической архитектуре (EWTA) набора доменов (предметных областей), которые объединяют группы связанных между собой технологий и компонентов. Каждый домен технологической архитектуры включает описание принципов, технологий, стандартов, продуктов, конфигураций, лучших практик, которые являются многократно используемыми строительными блоками при построении ИТ-систем. Таким образом, **документ, описывающий домен технологической архитектуры**, включает:

- формулировку миссии и стратегических целей домена;
- описание компонентов домена с точки зрения единого представления включенных в домен технологий;
- принципы проектирования, принятые в домене. Они определяют правила, применяемые в процессе принятия решений в отношении технологий домена;
- обоснования и последствия принятия принципов. Здесь могут быть построены матрицы соответствия между требованиями к

технологической архитектуре (RTA), сформулированными в процессе разработки, и принципами проектирования, принятыми для конкретного домена;

- продукты и технические стандарты, обеспечивающие выполнение требований к технологической архитектуре. Выделяют стратегические (предпочтительные) стандарты, переходные (временно используемые), устаревшие (подлежащие обновлению либо замене в ближайшее время) и исследовательские или новые (находящиеся на этапе рассмотрения либо апробации);
- примеры лучших практических реализаций;
- конфигурации, формулируемые в случаях, когда необходимо облегчить принятие сложных решений или уменьшить общую стоимость владения за счет стандартных конфигураций;
- несоответствия между существующим состоянием домена технологической архитектуры и желаемым состоянием. Это служит основой для последующей работы группы, которая отвечает за данный домен архитектуры.

Взгляд на технологическую архитектуру с точки зрения предоставляемых ею инфраструктурных сервисов обусловлен распространением принципов *сервис-ориентированной архитектуры*. Это связано с описанием, например, сервисов презентации информации (порталов, настольных систем и пр.), сетевых сервисов (LAN, WAN, удаленного доступа), сервисов безопасности (управления пользователями, доступа), сервисов хранения данных (SAN — Storage Area Network, файловые системы), сервисов баз данных (OLTP — Online Transaction Processing), интеграционных сервисов, платформенных сервисов, которые используются прикладными системами [1]. При этом по мере повышения уровня абстракции выделяется четыре группы сервисов:

1) базовые инфраструктурные сервисы: общие, стандартные технологии, широко используемые в рамках всех ИТ-систем предприятия. Они ориентированы не на разработчиков прикладных систем, а на специалистов по инфраструктуре.

Примерами являются ПО пересылки сообщений промежуточного слоя, мониторы транзакций, сервисы каталогов;

2) общие (framework) инфраструктурные сервисы: общие, совместно используемые технологии, не содержащие готовой бизнес-логики (хотя она и может быть запрограммирована), ориентированные на разработчиков и частично стандартизированные.

Примерами таких сервисов являются управление контентом, серверы приложений, серверы выполнения бизнес-правил;

3) общие (framework) бизнес-сервисы, которые могут быть использованы в рамках различных бизнес-процессов, поскольку содержат готовую, предопределенную бизнес-логику. Примерами таких сервисов являются модули определения цены товара, персонализации информации, оценки кредитного рейтинга;

4) прикладные бизнес-сервисы, содержащие специфическую для отдельных бизнес-процессов высокоуровневую бизнес-логику (сервисы CRM-систем или систем управления поставками).

Этап 3 — разработка плана реализации, обеспечивающего миграцию в сторону желаемого состояния архитектуры.

В полном описании методики META Group приводятся следующие разделы:

- практическая реализация архитектуры через процесс управления корпоративными ИТ-программами и проектами;
- вопросы управления и контроля архитектурного процесса;
- оценка зрелости архитектуры;
- анализ технологических тенденций и планирование;
- управление портфелем ИТ-активов и проектов.

Подробное описание методики разработки архитектуры предприятия META Group содержится в документе Enterprise Architecture Desk Reference.

4.4. Методика TOGAF

Методика описания архитектуры **TOGAF** (The Open Group Architecture Framework) [30] была предложена некоммерческим объединением The Open Group, которое входит ряд ведущих производителей информационных технологий, а также компаний

из списка «Fortune 1000»¹. TOGAF позиционируется ее авторами не как некоторая эталонная модель, а как «средство для разработки архитектур информационных систем». Основное назначение TOGAF — ускорить и облегчить процесс разработки архитектуры конкретной организации, обеспечивая при этом возможность будущего развития. В феврале 2009 года была опубликована **девятая версия** этой модели.

Основным полем для применения TOGAF является, прежде всего, программная инфраструктура ИС (в противоположность таким типам архитектур, как бизнес-архитектура, архитектура данных и приложений). Таким образом, данная методика в наилучшей мере подходит для описания интеграционных компонентов, используемых для поддержки широкого спектра корпоративных приложений, прежде всего, критичных для бизнеса. Поскольку интеграционная архитектура сильно зависит от принимаемых решений в остальных областях, то в рамках методики TOGAF в необходимой степени рассматриваются и эти смежные области.

В состав модели TOGAF входят два основных компонента: **методика ADM** (Architecture Development Method), определяющая процесс разработки архитектуры, и **базовая архитектура FA** (Foundation Architecture). Базовая архитектура дополняется соответствующей базой данных ресурсов, включающей описания архитектурных принципов, примеров реализации, а также специализированный язык ADML. Процесс разработки архитектуры по методике ADM включает следующие фазы:

- подготовка — уточнение модели с учетом особенностей организации, определение принципов реализации проекта;
- фаза А — определение границ проекта, разработка общего представления архитектуры; утверждение плана работ и подхода руководством;
- фаза В — разработка бизнес-архитектуры предприятия;
- фаза С — разработка архитектуры данных и архитектуры приложений;

¹ Список самых крупных по уровню дохода компаний США по версии журнала «Fortune».

- фаза D — разработка технологической архитектуры;
- фаза E — проверка возможности реализации предложенных решений;
- фаза F — планирование перехода к новой системе;
- фаза G — формирование системы управления преобразованиями;
- фаза H — управление изменением архитектуры.

Каждая фаза, в свою очередь, разбивается на подпроцессы (этапы), отдельные работы и так далее. Например, фаза D включает следующие основные подпроцессы:

- описание существующей технологической архитектуры:
 - обзор бизнес-архитектуры, архитектуры данных и приложений для определения начальных данных и необходимой степени детализации;
 - описание существующей системы с необходимой степенью детализации для выявления требуемых изменений при формировании целевой архитектуры. Создание реестра используемых платформ программного и аппаратного обеспечения;
 - выявление и описание элементарных архитектурных блоков — кандидатов на использование в новой архитектуре. Фактически, речь идет о возможных архитектурных шаблонах;
 - разработку черновика технического отчета, резюмирующего основные результаты изучения существующего состояния и возможности использования типовых блоков;
 - направление черновика отчета на рецензирование, анализ комментариев и внесение, при необходимости, поправок;
- формирование целевой технологической архитектуры:
 - описание существующей системы в терминах TOGAF;
 - определение перспектив (представлений) архитектуры;
 - формирование модели целевой архитектуры;
 - определение ИТ-служб (сервисов);
 - подтверждение учета бизнес-требований;
 - определение архитектуры и используемых блоков (шаблонов);
 - проведение анализа расхождений (gap analysis).

Для каждого такого подпроцесса определяются решаемые в ходе его выполнения задачи, входные и выходные документы. Важно отметить, что процесс предусматривает не обязательную, но возможную адаптацию метода к условиям конкретного предприятия, которая осуществляется на предварительной фазе. Это может быть вызвано как необходимостью учета других существующих стандартов предприятия, так и привлечением аутсорсинговых компаний к разработке архитектуры. Примером может являться проект внедрения корпоративной ERP-системы. В этом случае необходимо определенное изменение порядка разработки: поскольку бизнес-архитектура может определяться возможностями, поддерживаемыми в выбранном продукте, фазы В и С в данном случае будут выполняться не до, а после фазы D.

Процесс разработки не заканчивается выбором оптимальной архитектуры и разработкой плана миграции. Необходимыми элементами являются задачи, выполняемые на фазах G и H. В частности, для обеспечения практического принятия архитектуры в организации и успеха проекта обязательным является формирование системы управления реализацией архитектуры (*Implementation Governance*).

Важно отметить, что TOGAF распространяется свободно и может быть использована бесплатно любой организацией для разработки внутренних проектов. Лицензируется только коммерческое использование. В настоящее время с сайта The Open Group было загружено 90000 копий TOGAF. Количество проданных экземпляров серии TOGAF составило более 20000.

4.5. NASCIO¹ Architecture Toolkit

Набор шаблонов **IT Architecture Toolkit** первоначально позиционировался как специализированное средство для документирования ИТ-архитектуры организации.

¹ NASCIO (National Association of State Information Resource Executives) — Национальная ассоциация руководителей государственных информационных ресурсов (США, 1969 г.).

Основное преимущество использования набора шаблонов **IT Architecture Toolkit** заключается в построении иерархической системы описаний элементов, удобной для поддержания жизненного цикла документа, т. е. в форме, предполагающей его возможные изменения в будущем по мере изменения требований бизнеса и совершенствования технологий.

В версии 2.0 данного подхода структурная схема методики включает пять уровней [1]:

- 1) области, или домены, ИТ-архитектуры;
- 2) дисциплины;
- 3) технологические дисциплины;
- 4) продуктовые компоненты;
- 5) документы соответствия.

Области (домены) являются логическими блоками технологической архитектуры. Каждая область может включать одну и более дисциплин. Вся ИТ-архитектура подразделяется на набор областей верхнего уровня (доменов), описывающих отдельные аспекты ИТ-систем. В составе списка доменов предлагается выделять следующие области:

- управление приложениями;
- управление данными;
- управление информацией;
- интеграцию;
- управление пользователями и доступ;
- сети и коммуникации;
- платформы;
- управление системами;
- информационную безопасность и т. п.

Дисциплины обеспечивают логическое деление доменов на разделы, которыми уже проще управлять, т. е. домены включают в себя несколько функциональных дисциплин. Дисциплины представляют собой связанные единицы в рамках соответствующей предметной области. Каждая дисциплина содержит одну и более технологических дисциплин. Например, в домен «Управление системами» входят следующие дисциплины:

- управление активами (Asset management);
- управление изменениями (Change management);
- управление событиями (Event Management);
- поддержка пользователей (HelpDesk);
- обеспечение непрерывности бизнеса (Business continuity) и др.

Технологические дисциплины — это дисциплины, поддерживающие функциональные технологические разделы архитектуры. В качестве примера можно привести дисциплину «Управление данными» (Data Management), которая является частью области «Информация». Дисциплина «Управление данными» может включать такие технологические области:

- реляционные СУБД;
- плоские файловые системы;
- настольные базы данных;
- модели данных.

Каждая из технологических дисциплин включает свои продукты, протоколы и связанные с ними конфигурации. Это детализируется на уровне «Продуктовые компоненты». С указанного уровня начинается описание технических деталей технологической архитектуры.

Продуктовые компоненты включают протоколы, продукты (семейства продуктов) и конфигурации, которые специфичны для каждой технологической области. Примерами продуктовых компонентов, которые могут быть идентифицированы в рамках технологической области «Модели Данных», являются такие продукты, как ERWin, Visio и Designer 2000. Документация для каждого компонента включает оценочные критерии, которые были использованы для включения продуктового компонента в общую технологическую архитектуру.

Документы соответствия определяют руководства, стандарты и регулирующие документы, которые связаны с дисциплинами, технологическими дисциплинами и/или продуктовыми компонентами. Они предписывают необходимость соблюдения тех или иных международных рекомендаций, стандартов, законодательных актов. Документы соответствия могут присутство-

вать на каждом из пяти уровней. Они обеспечивают основу для принятия важных решений о новых продуктах, протоколах, конфигурациях и т. д.

Важным преимуществом такого подхода является возможность представления всего описания архитектуры в виде единой «гипертекстовой» базы данных, что позволяет эффективно организовать процессы управления жизненным циклом отдельных документов, а также эффективно разграничить права доступа к отдельным разделам (например, документам, описывающим применяемые средства защиты информации) при сохранении целостности и единства описания.

В табл. 4.3. приведен пример иерархии описания архитектуры в соответствии с рекомендациями NASCIO.

Таблица 4.3

Пример иерархии описания архитектуры
в соответствии с рекомендациями NASCIO

Область (домен)	Дисциплина	Технологическая дисциплина	Продуктовые компоненты	Документы соответствия
Информация	Управление данными	Реляционные СУБД	MS SQL Oracle DB2	Стандарты предприятия по именованию хранимых процедур
		Плоские файловые системы		Квоты на использование общего дискового пространства
		Настольные БД	MS Access	Стандарты предприятия по защите БД Access
		Модели данных	ERWin MS Visio Designer 2000	Стандарты предприятия по именованию таблиц и атрибутов

В табл. 4.4 дан пример модели технологической архитектуры, включающей в себя девять областей, которые, в свою очередь, разбиты на технические функциональные элементы или

дисциплины. Каждая организация должна определять свой набор технических дисциплин в зависимости от потребностей.

В версии 3.0 [31], опубликованной в октябре 2004 г., предмет рассмотрения охватывает уже и область бизнес-архитектуры, так что набор шаблонов может рассматриваться наряду с другими универсальными рамочными моделями.

Таблица 4.4

Области и дисциплины модели технологической архитектуры

Область	Дисциплины
Информация	Управление данными (Data Management). Управление знаниями. Геоинформационные системы (GIS). Хранение данных
Приложения	Управление средствами разработки приложений. Электронные средства совместной работы
Интеграция	Функциональная интеграция. Программное обеспечение промежуточного слоя (связующее ПО)
Доступ	Доступ. Branding: например, рекомендации по внешнему виду web-сайта госорганизации. Доступность
Сеть	Физическая сеть. Управление сетью
Платформа	Платформа: аппаратное обеспечение (серверы, настольные системы, системы хранения). Управление конфигурациями: стандарты на операционные системы, утилиты, конфигурации аппаратного обеспечения
Системное управление	Управление активами. Управление изменениями. Управление событиями. Управление инцидентами и проблемами. Непрерывность бизнеса (Business Continuity)
Частная информация	Профилирование. Персонифицирование. Обеспечение защиты частной информации
Безопасность	Корпоративная безопасность. Безопасность серверов

Набор шаблонов версии 3.0 включает пять взаимосвязанных архитектур:

- 1) архитектуру управления;

- 2) бизнес-архитектуру;
- 3) информационную архитектуру;
- 4) технологическую архитектуру (*практически соответствует всей ИТ-архитектуре, рассмотренной в версии 2*);
- 5) архитектуру решений (Solution Architecture), обусловившую частичное изменение структуры модели.

Шаблоны «**Архитектура управления**» интересны специалистам, занимающимся процессом управления архитектурой предприятия или планирующим в будущем посвятить себя такой деятельностью. В данной архитектуре наряду с описанием управления персоналом, ролями и обязанностями предлагаются инструменты оценки и типовые структуры организаций.

В составе **бизнес-архитектуры** предлагается выделить несколько бизнес-областей (доменов). Это разделение может производиться как по функциональному (например, образование, здравоохранение, социальное обеспечение), так и по некоторому «тематическому» (например, услуги гражданам, взаимодействие с другими органами власти, внутренние процессы) или географическому признаку.

В составе бизнес-областей выделяются отдельные архитектурные компоненты, рассмотрение которых может вестись с различных перспектив и фокусов модели Захмана (на двух верхних уровнях этой модели). Такие компоненты модели Захмана, как «Кто» и «Зачем» могут быть объединены в один общий — «Стратегический бизнес». Важно отметить, что одни и те же выбранные перспективы и фокусы должны будут применяться ко всем бизнес-областям.

Для каждой бизнес-области в описании определяются существенные принципы и существующие тенденции. Далее в составе бизнес-области формируются подчиненные документы, описывающие ее компоненты, которые фактически соответствуют отдельным ячейкам в верхних двух уровнях модели Захмана.

Таким образом, в состав компонентов могут входить, например, описание ролей (должностей) и их ответственности в организации, важные с точки зрения предприятия события и циклы деятельности, расположения офисов и т. п. Важным атрибутом

описания является индикация типа состояния компонента, т. е. принадлежности к существующему или целевому состоянию.

Информационная архитектура включает архитектуру данных и архитектуру процессов. Построение данной архитектуры облегчает управление информацией предприятия, обеспечивая упорядочение деловых отношений и выявление сущности принятых предприятием деловых правил, а также позволяет улучшить организацию бизнес-процессов в соответствии с возможностями поддерживающих их информационных систем.

Технологическая архитектура представляет типовые модели процессов, шаблоны для документации технологии и критериев согласия в пределах организации. Эти шаблоны включают типовые инструменты, данные и сообщения относительно архитектур, заимствованных у различных организаций с успешными реализациями программ архитектуры предприятий.

Новым понятием в наборе шаблонов IT Architecture Toolkit является **архитектура решений**, облегчающая описание ее элементов в пределах предприятия посредством определенного перечня предлагаемых архитектору направлений при формулировании требований к решениям, спецификации проекта и логическим моделям проекта. Фактически данная архитектура представляет слой шаблонов из модели Gartner (см. параграф 4.2). В отличие от остальных архитектур, описания элементов архитектуры решений не содержат такого признака, как «существующий» или «целевой». Фактически все эти «решения» относятся как раз к стадии перехода от существующего состояния к целевому, так что по завершении соответствующего проекта они сохраняются со статусом «архивный» [1]. Для описания конкретного решения используются три типа шаблонов:

1) «область» — тип шаблона, описывающий полное решение и связь решения с планом выполнения. Данный шаблон определяет концептуальную модель решения;

2) «требования» — тип шаблона, содержащий формализованные требования к наборам решений, сгруппированных по типам. Организация шаблона схожа с ГОСТ 34.698-9 «Техническое задание на АС»;

3) «дизайн» — тип шаблона, предоставляющий списки различных спецификаций проекта, основанные на определенных типах, представлениях и категориях. Эти шаблоны содержат информацию, позволяющую оценить воздействия решений на состояние внешней среды в различных областях на определенный момент времени.

Наряду с описанием элементов архитектуры в ходе процесса разработки определяется реализация стандартных процессов поддержки жизненного цикла архитектуры применительно к конкретным особенностям предприятия. К таким процессам, в частности, относятся:

- документирование;
- рецензирование;
- информирование;
- изменение;
- проверка соответствия;
- поддержка актуальности;
- организация и управление разработкой архитектуры.

Диаграммы процессов строятся применительно к набору типовых предопределенных ролей (например, Рецензент, Документатор, Лидер и т. п.), которые присваиваются отдельным сотрудникам, должностям или подразделениям. Эти роли определяют права и ответственность данных участников процесса.

Данный подход использован в практике построения архитектуры предприятия во многих американских штатах и федеральных организациях.

4.6. Модель представления архитектуры «4+1»

Достаточно важную роль в развитии подходов к описанию архитектуры предприятия сыграла модель «4+1» (The 4+1 View Model of Architecture), предложенная Ф. Кручтенем (Philippe Kruchten) из компании Rational в 1995 г.

Данная методика позиционировалась, прежде всего, как способ описания архитектуры систем, основанных на активном использовании программного обеспечения, хотя идеи, заложенные в эту методику, могут использоваться и в более широком контексте архитектуры предприятия. Модель предлагает простой и понятный способ описания архитектуры сложных систем, который состоит в использовании четырех основных представлений и объединяющего их сценария (рис. 4.4) [32].

Представлениями в этой модели являются:

- 1) логическое представление;
- 2) процессное представление;
- 3) физическое представление;
- 4) представление уровня разработки.

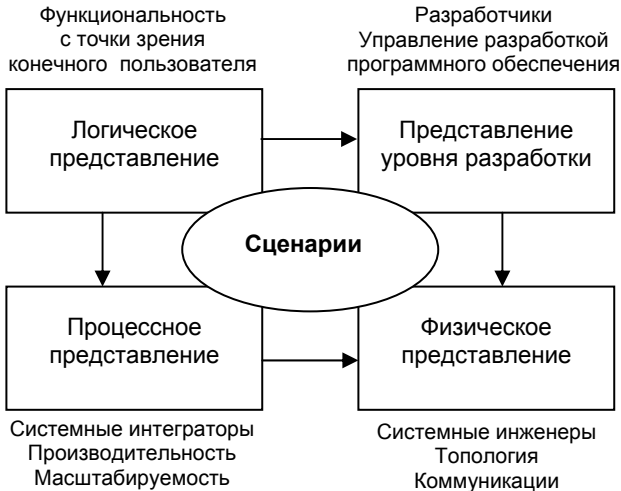


Рис. 4.4. Модель «4+1»

Логическое представление является объектной моделью проектирования в том случае, если используется объектно-ориентированная модель проектирования. Основной целью логического представления в данной модели является описание функциональных требований (назначение системы в терминах конечных поль-

зователей). Для этого представления используются различные абстрактные конструкции, такие как объекты и классы объектов. Для их иллюстрирования могут применяться диаграммы классов в нотации языка UML либо, например, диаграммы «сущность-связь», если в разработке приложения доминируют данные.

Процессное представление описывает вопросы параллельного исполнения и синхронизации процессов. В нем учитываются некоторые нефункциональные требования к системе, включающие производительность и доступность. С помощью этого представления рассматриваются такие аспекты, как одновременное выполнение и распределение процессов, интеграция системы, устойчивость к сбоям, а также соответствие основных объектов абстракции, рассмотренных на уровне логического представления, архитектуре процессов. Архитектура процессов может быть представлена на различных уровнях абстракции. На высшем уровне система рассматривается как набор независимо выполняемых сетей взаимодействующих между собой программ. На более низких уровнях рассматриваются процессы и задачи.

Физическое представление описывает размещение программных компонентов системы на аппаратных платформах и аспекты, связанные с физическим расположением системы. В основном рассматриваются нефункциональные требования, такие как доступность, надежность, устойчивость, производительность, масштабируемость. Этот уровень описывает распределение элементов — сетей, процессов, задач и объектов — по различным узлам (элементам аппаратного обеспечения, объединенным в сеть).

Представление уровня разработки описывает статическую организацию программной системы в среде разработки, фактическую организацию модулей системы, разделение ее на подсистемы, которые могут разрабатываться независимо.

Описание архитектуры системы посредством четырех основных представлений иллюстрируется и проверяется путем использования **сценария**, сформированного при объединении этих представлений. Архитектура системы во многом определяется данными сценариями. Каждое представление отражает специфические аспекты моделируемой системы. Сценарии описываются

как последовательность взаимодействия объектов и процессов. Они отражают наиболее существенные требования, которым должна удовлетворять система. Использование сценариев позволяет:

- идентифицировать элементы архитектуры, которые требуются для эффективно работающей системы;
- выполнять проверку работоспособности и полноты архитектуры;
- предоставлять иллюстрацию процесса построения архитектуры;
- формировать основу для проведения тестирования архитектурного прототипа.

4.7. Стратегическая модель архитектуры SAM

Стратегическая модель архитектуры SAM (Strategic Architecture Model) является инструментом анализа и документирования архитектуры предприятия и связанных с ней доменов. Методика создания модели разработана английской консалтинговой компанией Systems Advisers Ltd [33]. SAM использует нотацию «сфер интересов» для представления целостного набора фактов о предприятии и «отношений», которые связывают эти факты в полезные группы, что обеспечивает полезный взгляд на структуру и операции, выполняемые предприятием. SAM можно рассматривать как некоторую надстройку над моделью архитектуры предприятия Захмана. Она предоставляет общие структуры для определения архитектуры и механизмы, позволяющие осуществлять организацию и анализ информации об архитектуре. SAM использует итеративный подход для создания архитектуры, сочетающий элементы разработки «сверху–вниз» и «снизу–вверх». «Сферы интересов» SAM позволяют легко систематизировать всю информацию, имеющую отношение к определенному предмету, например информацию об организационных структурах или бизнес-процессах.

Сфера может заполняться в направлении «снизу–вверх» путем сбора относящейся к предметной области информации, которая обобщается на более высоких уровнях. Либо же заполнение может идти в направлении «сверху–вниз» с постепенной декомпозицией на более мелкие детали.

После того как некоторая пара сфер определена с достаточной степенью детализации, элементы, составляющие эти сферы, могут быть связаны так, чтобы представить существующие в реальности связи между объектами анализа. Это обеспечивает возможность оптимизации и улучшений в различных областях деятельности предприятия. Наибольшую важность представляют следующие сферы: цели и задачи, организация, бизнес-процессы, прикладные системы, технологии, проекты, бизнес-компоненты, данные, бизнес-функции, инфраструктура (рис. 4.5).

Внутри каждой области интересов сохраняется информация об определенной предметной области, что обеспечивает простоту ее сопровождения и извлечения. Обычно применяется одна или несколько иерархических структур, которые напоминают ящик с файлами. Минимальный объем информации, относящейся к какой-либо сфере, называется элементом (member). Например, элементами сферы «Местоположение» могут быть «Головной офис», «Офис продаж в Санкт-Петербурге», «Завод в Томске». Другими примерами элементов различных сфер являются:

- конкретные подразделения (например, «Отдел продаж» в сфере «Организация»);
- конкретные бизнес-процессы (например, «Прием заказа» в сфере «Бизнес-процессы»);
- конкретные информационные объекты (например, «Клиент» в сфере «Данные»).

Группировка функций, создающих и обновляющих одни и те же элементы данных с помощью процесса «Коммутативная кластеризация», позволяет определить избыточное количество «строительных блоков» — компонентов, которые могут использоваться для построения систем и приложений, поддерживающих определенные бизнес-процессы.

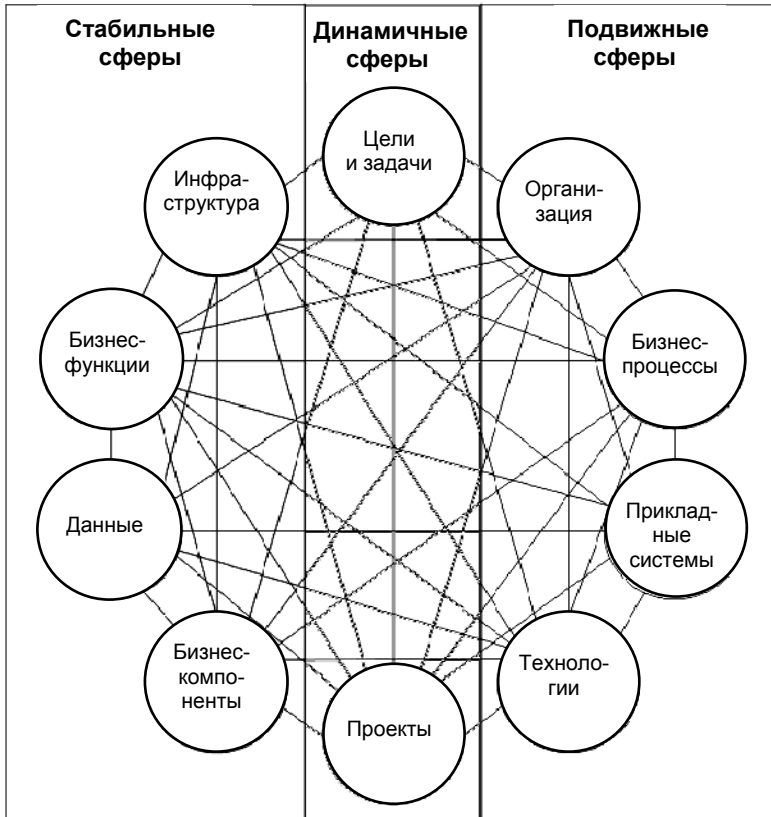


Рис. 4.5. Типичные сферы интересов SAM

Компоненты являются важными конструкциями в современных подходах к разработке систем. Они предлагают «сервисы», которые могут использоваться в совокупности с сервисами, предлагаемыми другими компонентами в рамках сервис-ориентированной архитектуры.

Можно выделить три категории сфер (см. рис. 4.5):

1) **стабильные**, описывающие достаточно стабильные элементы бизнеса и представляющие фундаментальные структуры: бизнес-функции, данные, бизнес-компоненты и инфраструктуру;

2) *подвижные*, описывающие действия предприятия с точки зрения организации бизнеса в настоящем и будущем с целью создания отличий от конкурентов и обеспечения динамичности в деятельности. Сферы этой категории — организация, бизнес-процессы, прикладные системы и технологии — представляют собой области, которые организация может изменить достаточно быстро. Эти сферы могут находиться в процессе постоянных изменений, обеспечивая адекватную реакцию на изменение экономических и рыночных условий;

3) *динамичные*, задающие направления бизнеса, рабочие программы, управление изменениями. Они описывают основные области, в которых работает предприятие, и усилия, направленные на достижение целей бизнеса посредством реализации связанных между собой проектов.

Данная классификация позволяет:

- понимать, какая часть архитектуры предприятия носит достаточно стабильный характер, а какая требует постоянных изменений;
- идентифицировать относительно стабильные области, для которых полезна разработка архитектурных шаблонов. Такими областями являются бизнес-функции, данные, бизнес-компоненты и в определенной степени инфраструктура.

4.8. Архитектурные концепции и методики Microsoft

Крупные компании-поставщики инфраструктурных информационных технологий, такие как Microsoft, IBM, SAP и другие, могут «позволить себе роскошь» создания собственных методик разработки архитектуры ИС предприятия с учетом своей области специализации. В то же время это является в какой-то степени и обязанностью таких компаний, поскольку спектр предлагаемых ими технологий покрывает существенную часть архитектуры предприятия в целом и специалистам нужны соответствующие

практические рекомендации непосредственно от поставщиков. Взгляды компании Microsoft [34] в большей степени сфокусированы на процессах разработки конкретных программных прикладных систем и создании технологической инфраструктуры, включая центры обработки данных различного масштаба и уровня надежности.

Как и во многих других методиках, здесь выделяются четыре представления (домена):

- 1) бизнес-архитектура;
- 2) архитектура информации;
- 3) прикладные системы;
- 4) технологическая архитектура.

Данные представления рассматриваются на различных уровнях абстракции: *концептуальном, логическом и физическом*. Кроме того, явно выделяются процессы разработки прикладных систем, организация процессов эксплуатации технологической инфраструктуры и создание соответствующих шаблонов, которые могут использоваться как при проектировании архитектуры систем, так и при ее построении. При этом компания Microsoft выработала достаточно подробные методики, отражающие различные аспекты архитектуры и, прежде всего, процессы разработки систем и создания инфраструктуры и процессы эксплуатации систем и инфраструктуры.

К таким методикам относятся, в частности, четыре взаимодополняющие методики: Microsoft Solutions Framework (MSF), Microsoft Operations Framework (MOF), Microsoft Systems Architecture (MSA) и Microsoft Solutions for Management (MSM), которые содержат рекомендации для специалистов, касающиеся следующих основных вопросов:

- 1) как правильно создавать ИТ-системы (MSF);
- 2) как правильно создавать технологическую инфраструктуру (MSA);
- 3) как правильно эксплуатировать технологическую инфраструктуру (MOF);
- 4) как правильно строить процессы управления технологической инфраструктурой (MSM).

Методики MSF и MSA в большей степени относятся к процессу разработки архитектуры прикладных систем и инфраструктуры соответственно, а методики MOF и MSM — к архитектуре системного управления, т. е. к вопросам управления и эксплуатации. При этом MOF и MSF нацелены на различные, но связанные между собой фазы жизненного цикла ИТ-решений (рис. 4.6).

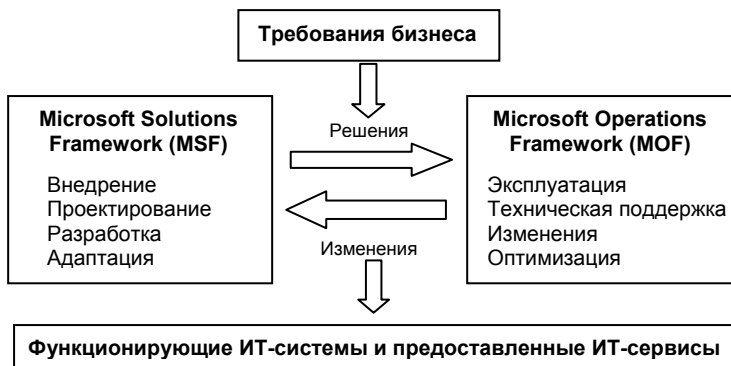


Рис. 4.6. Организация взаимодействия MSF и MOF для удовлетворения запросов бизнеса

Методики Microsoft сосредоточены в основном на системном уровне — уровне архитектуры прикладных систем и обеспечивающей инфраструктуры. В этой достаточно «узкой» области полезными являются приведенные соотношения между различными перспективами описания системы и моделями, используемыми для описания на соответствующем уровне абстракции (рис. 4.7). В идеале для каждой перспективы используется какой-то один тип моделей так, как это показано на рисунке. Но в реальности могут использоваться и несколько различных моделей для описания каждой из перспектив, т. е. концептуальной, логической и физической архитектур системы.

На рис. 4.8 показаны взаимосвязи между различными перспективами в описании архитектуры и используемыми шаблонами проектирования, а также примерно отображено соответствие между методиками Microsoft и элементами архитектуры предприятия.

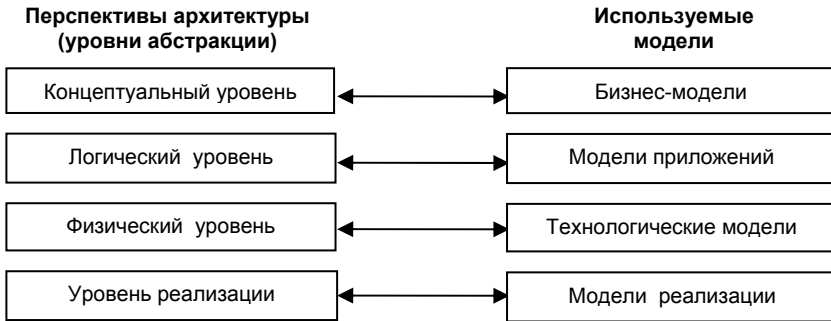


Рис. 4.7. Различные перспективы архитектуры системы и используемые модели

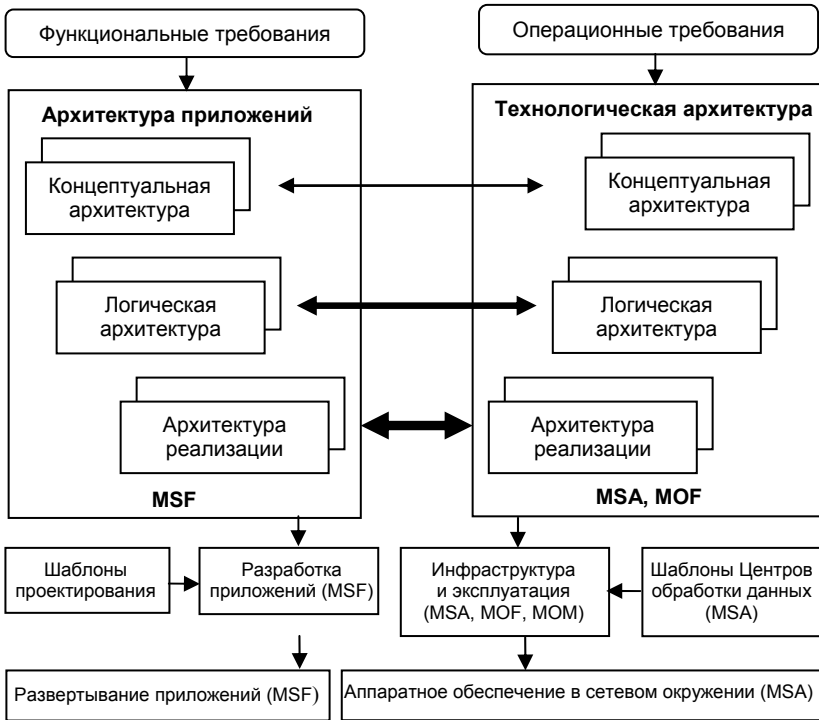


Рис. 4.8. Организация взаимосвязи компонентов архитектуры предприятия с использованием методике Microsoft

Microsoft выделяет два типа руководств и обеспечивающих методик, которые могут помочь системным архитекторам ускорить процессы разработки моделей при минимизации рисков.

Первый тип руководств — это архитектурные концепции, такие, например, как сервис-ориентированные подходы к проектированию архитектуры.

Данные концепции обеспечивают:

- общее понимание и язык описания архитектуры;
- общие руководства, рекомендации по использованию специфических концепций;
- указания способов реализации концепции на практике в форме конкретных технологий и стандартов.

Второй тип руководств, которыми могут пользоваться системные архитекторы, — это архитектурные шаблоны, основанные на практическом опыте большого количества успешно реализованных проектов создания распределенных прикладных систем и являющиеся следствием использования вышеописанных архитектурных концепций.

Эти шаблоны содержат в себе лучшие практики проектирования распределенных приложений и средства по минимизации рисков неудач проектов, поскольку рекомендуют хорошо апробированные модели.

Типы руководств — архитектурные концепции и шаблоны — могут присутствовать и использоваться на различных уровнях проектирования архитектуры прикладной системы (рис. 4.9).

Уровнями проектирования архитектуры прикладной системы являются:

- уровень концептуальной архитектуры в форме концепций построения бизнес-моделей и соответствующих шаблонов;
- уровень логической архитектуры в форме концепций построения моделей приложений и соответствующих шаблонов;
- уровень физической архитектуры в форме концепций построения технологических моделей и соответствующих шаблонов.

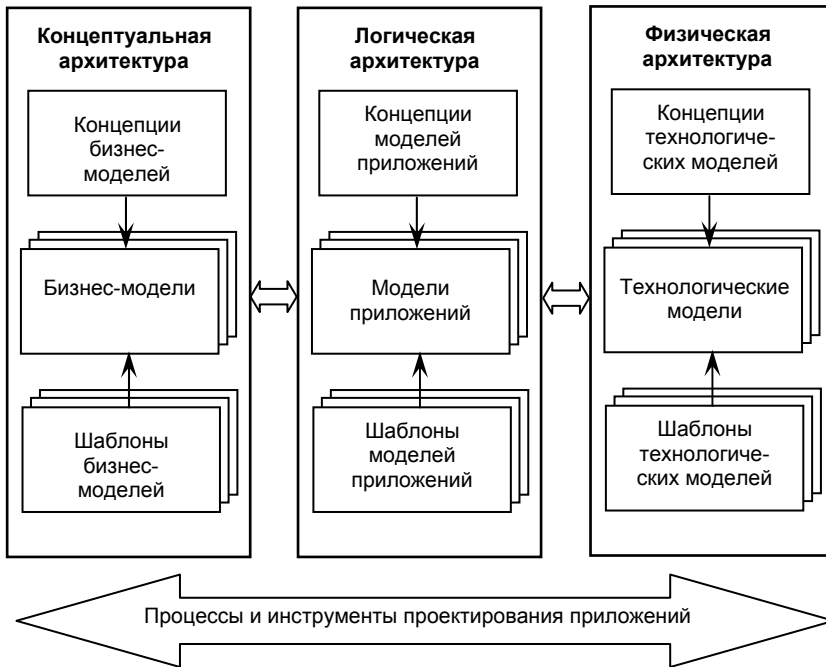


Рис. 4.9. Концепции и шаблоны по построению архитектуры приложений

Использование данных концепций и шаблонов является важным условием успешного, быстрого и эффективного с точки зрения затрат создания систем и использования ИТ-технологий организациями. Поэтому, помимо методик MSF, MOF, MSA и MSM, компанией опубликованы подробные руководства по разработке архитектуры систем, а также шаблоны, которые могут применяться при проектировании корпоративных ИС¹.

¹ Данные документы можно найти в открытом доступе на web-страницах Microsoft:

<http://msdn.microsoft.com/architecture;>

<http://msdn.microsoft.com/practices;>

<http://www.microsoft.com/resources/practices;>

<http://www.microsoft.com/systemsarchitecture.>

4.9. Метод планирования архитектуры организации ЕАР

ЕАР (Enterprise Architecture Planning)¹ представляет собой метод планирования архитектуры предприятия, основу которого составляют:

- процесс планирования архитектуры предприятия, ориентированный на создание архитектуры, обеспечивающей поддержку бизнеса на основе учета конкретных данных, приложений и технологий, наиболее полно отвечающих потребностям предприятия;
- разработка плана реализации, определяющего процесс воплощения этой архитектуры. При этом предполагается, что созданию архитектуры предшествует разработка бизнес-стратегии, включающей миссию, бизнес-цели и способы их достижения.

ЕАР декларирует 10 этапов (табл. 4.5), определяющих состав и структуру слоев и элементов архитектуры, а также план ее проектирования, обеспечивающий реализацию как традиционных требований к архитектуре, так и специфических требований конкретной организации. Этапы организованы в виде четырехуровневой схемы метода ЕАР:

- 1) уровень 1 (исходная позиция) — выработка решений, которые необходимо принять для реализации соответствующей архитектуры организации, и определение состава необходимого для реализации инструментария;
- 2) уровень 2 (анализ текущего состояния) — определение точки отсчета для преобразования существующей архитектуры в целевую, а также формирование временного графика перехода;
- 3) уровень 3 (планируемая перспектива) — определение технических деталей перспективной архитектуры (данные, приложения и технологии);
- 4) уровень 4 — формирование плана реализации перспективной архитектуры.

¹ Разработан Стивеном Спиваком [35].

Таблица 4.5

Этапы планирования архитектуры

Название этапа	Результаты
1. Инициация планирования	Цели, видение, методологии, инструментарий, команда, презентации, рабочий план
2. Предварительное бизнес-моделирование	Организационно-штатная структура, предварительная функциональная бизнес-модель
3. Формирование снимка организации	Полная функциональная бизнес-модель
4. Описание текущих систем и технологий	Каталог информационных ресурсов, системные схемы
5. Формирование архитектуры данных	Определения сущностей, ER-модель, матрица сущности-функции, отчет по архитектуре данных
6. Формирование архитектуры приложений	Определения приложений, матрицы приложений, анализ покрытия, отчет по архитектуре приложений
7. Формирование технологической архитектуры	Распределение данных/приложений, отчет по технологической архитектуре
8. Разработка плана реализации	Последовательность, план перехода, цены и преимущества, факторы успеха и рекомендации
9. Заключительное планирование	Окончательный отчет, презентация
10. Переход к реализации	Совершенствование политик, стандартов, процедур, детализация проектных планов

Этап «**Инициация планирования**» включает в себя 7 шагов, цели, задачи и основные результаты которых описаны ниже.

Шаг первый — формальное определение области и целей планирования архитектуры для понимания участниками проекта конечных достигаемых результатов. Итог данного шага представлен перечнем согласованных и утвержденных целей, а также списком причастных к проекту подразделений организации.

Основные задачи первого шага состоят в следующем:

- обзор организации и определение ее контекста (системных входов/выходов);
- оценка характеристик деятельности предприятия по степени их влияния на реализацию проекта: способствующие реализации и затрудняющие реализацию (например: несоответствие существующих информационных систем требованиям и высокая стоимость их сопровождения; необходимость в интеграции и распределении данных; наличие неуспешных ИТ-проектов по причинам ограниченности менеджмента по времени и бюджету и т. п.);
- формирование перечня строго сформулированных целей и их достижимости;
- формирование перечня подразделений, затрагиваемых грядущими изменениями ИТ-стратегии и корпоративной культуры.

Шаг второй — исследование организации, системных входов/выходов и вариантов представлений менеджеров. Результатом является согласованное и утвержденное видение организации, обеспечивающее политическую поддержку менеджмента.

Основными задачами шага являются:

- изучение всех исходных материалов по бизнесу (заказчики, продукты, сотрудники, цели и т. д.);
- определение лиц, заинтересованных в создании архитектуры;
- анализ опыта организаций, успешно выстроивших собственные архитектуры;
- формирование видения организации в разрезе ИТ-среды, обеспечивающей достижение целей.

Шаг третий — адаптация методологии планирования и создание руководства по ее реализации. Основными задачами шага являются:

- формулирование принципов и требований к методологии;
- оценка существующих на предприятии методов и стандартов;
- изучение имеющихся на рынке подходов;

- принятие решения об исполнителе (внутренние ресурсы или внешний консультант);
- создание методологии, отвечающей потребностям данного предприятия;
- разработка содержания отчетов, создаваемых на каждом из последующих этапов.

Шаг четвертый — инвентаризация компьютерных ресурсов и оценка инструментария создания архитектуры предприятия.

Основными задачами шага являются:

- определение требований к инструментарию;
- определение требований к аппаратуре;
- оценка альтернатив для репозитория проекта;
- выбор и приобретение подходящего программного инструментария;
- разработка регламентов и процедур, обеспечивающих надлежащее использование продуктов;
- разработка проектов отчетов, экранных форм и т. п.;
- оценка трудозатрат на «канцелярскую» поддержку большого объема документации по архитектуре предприятия;
- доведение решений по инструментарию до всех подразделений — потенциальных пользователей архитектуры.

Шаг пятый — создание проектной команды. Основными задачами шага являются:

- определение квалификационных требований по каждому этапу создания архитектуры;
- оценка трудозатрат по каждому этапу создания архитектуры;
- определение необходимого числа участников;
- спецификация ролей и областей ответственности каждого члена команды;
- подбор персонала;
- обучение персонала (методологии и инструментарий);
- выбор внешних консультантов и определение направлений их использования.

Шаги шестой и седьмой — подготовка рабочего плана, его презентация и утверждение. Основные задачи этих шагов традиционны, их рассмотрение выходит за рамки настоящей книги. В результате должен быть сформирован рабочий план и утвержден бюджет выполнения работ.

Задачей этапа **бизнес-моделирования** является обеспечение полной и исчерпывающей базой знаний всех участников проекта для ее использования при определении архитектуры и плана ее реализации. Бизнес-моделирование предполагает построение предварительной бизнес-модели, за которым следует построение полной бизнес-модели. Предварительная бизнес-модель идентифицирует функции и организационные единицы (исполнителей функций). По оценкам ряда экспертов этап «Предварительное бизнес-моделирование» требует 25–30 % всех трудозатрат на моделирование, он осуществляется в 3 шага.

Шаг 1 — документирование организационной структуры. В качестве результатов имеет обновленные организационные схемы, перечень ролей и мест их выполнения, оценку количества сотрудников по ролям. Основными задачами шага являются:

- формирование (редактирование) организационных схем и фиксация их в инструментарии;
- идентификация видов деятельности в разрезе организационных единиц;
- формирование отчетов по полученным результатам.

Шаг 2 — определение структуры бизнес-модели (идентификация и определение бизнес-функций). В качестве результатов имеет определенные функции, для которых определены следующие компоненты:

- имя;
 - краткое описание либо декомпозиция на подфункции;
 - принадлежность к конкретной организационной единице.
- Основными задачами шага являются:
- определение основных видов деятельности и бизнес-процессов;
 - функциональная декомпозиция процессов;

- развитие функциональной декомпозиции до уровня бизнес-операций;
- построение функционального иерархического дерева;
- оценка качества декомпозиции и ее улучшение;
- сопоставление функций и исполняющих их организационных единиц, построение соответствующей матрицы.

Шаг 3 — документирование бизнес-модели и ее распространение для верификации. Основными задачами шага являются:

- формирование отчетов по бизнес-модели;
- распространение отчетов и проведение презентации;
- сбор замечаний и предложений.

Полная функциональная бизнес-модель дает ответы на следующие вопросы:

- какая информация используется при выполнении функций;
- когда функция выполняется;
- где и кем функция выполняется;
- как часто функция выполняется;
- какие улучшения возможны.

Этап **«Формирование снимка организации»** включает в себя следующие 3 шага: 1) планирование, подготовку и проведение интервью; 2) построение бизнес-модели; 3) анализ бизнес-модели.

Планирование интервью включает: формирование списка интервьюируемых с указанием даты и времени проведения; распределение интервьюирующих по видам деятельности и бизнес-процессам (функциональным направлениям); подготовку инструкции для участников (цели и задачи, кто, когда, где, какие вопросы и др.); корректировку при необходимости плана создания архитектуры предприятия. Подготовка интервью включает разработку форм для управления процессом интервьюирования и фиксации результатов (прежде всего, для определения функций и информационных источников). Главная цель интервьюирования — выявление необходимых данных по бизнес-модели. На следующих шагах осуществляется обработка результатов интервью,

построение детальной модели, ее анализ, формирование пакета отчетов и проведение презентации.

Целью этапа «**Описание текущих систем и технологий**» является документирование всех используемых в организации системных и технологических платформ, т. е. создание так называемого каталога информационных ресурсов IRC (Information Resource Catalog), по-другому — системной энциклопедии, являющейся высокоуровневым объектом (а не детальным словарем данных).

Построение каталога информационных ресурсов включает четыре шага.

Шаг 1 — определение видов данных для IRC и проектирование форм для сбора данных. Основные задачи шага включают:

- определение видов данных по приложениям;
- определение видов данных по входам, выходам, файлам и БД приложений;
- идентификацию технологических платформ и определение их декомпозиции по видам (например, принтеры — матричные, лазерные; языки — Си, Паскаль и т. п.);
- проектирование форм для сбора данных;
- подготовку детальных инструкций по заполнению форм.

Шаг 2 — сбор данных для IRC и их ввод (заполнение форм), а также сопоставление приложений и функций. Основные задачи шага состоят в следующем:

- сбор системной документации;
- сопоставление приложений и бизнес-функций и формирование соответствующей матрицы;
- сопоставление приложений и технологических платформ и формирование соответствующей матрицы;
- ввод информации в инструментарий.

Шаг 3 — интегрирование и верификация информации по текущим приложениям и технологическим платформам, разработка потоковых диаграмм по каждой системе. Основные результаты — верифицированный IRC, пакет отчетов по IRC, предложения по улучшению IRC на основе проведенных обсуждений.

Шаг 4 — подготовка к процессам администрирования и сопровождения IRC, назначение которых состоит в поддержании каталога в актуальном состоянии. Здесь разрабатывается регламент поддержки, политики и процедуры сопровождения IRC, назначается ответственный по его сопровождению.

На этапе «**Формирование архитектуры данных**» определяются и идентифицируются основные разновидности данных, поддерживающих бизнес-функции. Архитектура данных представляется с помощью ER-модели. Этап содержит 4 шага:

- 1) формирование списка кандидатов в сущности (трудозатраты — 10 %);
- 2) определение сущностей, атрибутов и отношений (трудозатраты — 60 %);
- 3) сопоставление сущностей и бизнес-функций (трудозатраты — 20 %);
- 4) анализ результатов (трудозатраты — 10 %).

Целью **первого шага** является идентификация всех потенциальных сущностей, необходимых для поддержки бизнеса. Здесь осуществляется распределение компонентов бизнес-модели по членам команды в разрезе видов деятельности и бизнес-процессов), подготовка каждым из участников списка кандидатов, формирование общего списка кандидатов в сущности.

Целью **второго шага** является создание стандартного определения и описания каждой сущности, обеспечение графической иллюстрации их взаимодействия. Здесь сущности определяются и документируются, осуществляется построение ER-модели, производится сопоставление файлов и БД из IRC с сущностями.

Целью **третьего шага** является сопоставление сущностей с бизнес-функциями и приложениями, результатами которого являются матрица сущности-функции и матрица сущности-приложения. При этом для каждой функции нижнего уровня детализации идентифицируется вид каждой из затрагиваемых ей сущностей (создается, изменяется, используется), а приложения сопоставляются с сущностями по входам, выходам, файлам и БД.

Целью **четвертого шага** является подготовка, распространение и анализ отчета по архитектуре данных.

На этапе «**Формирование архитектуры приложений**» определяются основные виды приложений, необходимых для управления данными и поддержки бизнес-функций. Архитектура приложений не является ни системным проектом, ни набором детальных требований к системам. Она только определяет, какие приложения будут управлять данными, и снабжает соответствующей информацией исполнителей бизнес-функций.

Основными шагами этапа являются:

- 1) формирование списка потенциальных приложений (трудозатраты — 10 %);
- 2) определение приложений (трудозатраты — 50 %);
- 3) сопоставление приложений и функций (трудозатраты — 15 %);
- 4) анализ применимости существующих приложений (трудозатраты — 15 %);
- 5) анализ результатов (трудозатраты — 10 %).

Цель *первого шага* — идентификация каждого из возможных приложений и формирование их списка, при этом особое внимание уделяется приложениям, которые могут улучшить бизнес или обеспечить конкурентные преимущества.

Цель *второго шага* — снабжение каждого приложения стандартным описанием (определением) и построение графической схемы архитектуры приложений. Основные задачи шага:

- распределение приложений между членами команды;
- определение состава каждого приложения (имя, номер, цель, общее описание и возможности, бизнес-преимущества);
- упрощение сложных приложений и ликвидация избыточности;
- выработка предварительных предложений по применимости имеющегося на рынке ПО и технологических платформ;
- построение схемы архитектуры приложений;
- оценка качества архитектуры приложений (понимаемость, полнота и состоятельность, прочность-устойчивость).

Целью *третьего шага* является идентификация бизнес-функций, поддерживаемых или выполняемых приложениями. Здесь

для каждого приложения формируется матрица приложения-функции, а также перечень функций, не поддерживаемых ни одним приложением (с объяснением причин), а также матрица приложения, содержащего организационные единицы.

Цель **четвертого шага** состоит в определении соответствия архитектуры приложений существующим в организации приложениям. Здесь производится сопоставление каждого приложения из архитектуры приложений и имеющихся систем, определенных в IRC, а также контроль полноты сопоставления (каждое существующее приложение из IRC должно быть соотнесено, по крайней мере, с одним из архитектурных приложений). Кроме того, строится таблица соответствий архитектуры приложений и существующих приложений.

На **пятом шаге** производится подготовка, распространение и анализ отчета по архитектуре приложений.

Этап «**Формирование технологической архитектуры**» определяет основные виды технологий, необходимых для обеспечения окружения приложений, управляющих данными. Технологическая архитектура не является ни проектом сетевого оборудования и ПО, ни набором детальных требований к ним. Она только определяет виды технических платформ, поддерживающих бизнес. Основными шагами этапа являются:

1) идентификация технических принципов и платформ (трудозатраты — 15 %);

2) определение платформ и их распределение (трудозатраты — 50 %);

3) сопоставление платформ с приложениями и бизнес-функциями (трудозатраты — 20 %);

4) анализ результатов (трудозатраты — 15 %).

Целью **первого шага** является формулирование общих принципов для технических платформ и идентификация потенциальных кандидатов в платформы.

Цель **второго шага** — определение стратегии распределения приложений и данных, технологических платформ на основании сформулированных принципов. Основными результатами шага являются распределение данных и приложений, конфигурация

технологических платформ, оценка концептуальной архитектуры. Основными задачами шага являются:

- определение мест размещения бизнес-функций;
- распределение данных и приложений;
- определение конфигурации технологических платформ (рабочих станций, сети, архитектуры бизнес-систем);
- оценка концептуальной технологической архитектуры.

Цель *третьего шага* — обоснование технологических платформ путем их соотнесения с использующимися бизнес-функциями, формирование таблиц-платформ приложений бизнес-функций.

На *четвертом шаге* производится подготовка, распространение и анализ отчета по технологической архитектуре.

Этап «**Разработка плана реализации**» включает:

- 1) формирование последовательности реализации приложений;
- 2) оценку трудозатрат и ресурсов, построение плана;
- 3) оценку стоимости и достоинств плана;
- 4) определение факторов успеха и рекомендаций по их достижению.

Целью *первого шага* является установка приоритетов и формирование последовательности реализации приложений (например, приложения, порождающие данные, должны быть реализованы перед реализацией приложений, использующих эти данные). Основные результаты шага: матрица приложений, содержащих сущности данных, список упорядоченных по приоритетам приложений, план модификации и/или замены существующих систем, группировка приложений в проекты, последовательность реализации технологии.

Основными задачами первого шага являются:

- сопоставление приложений и сущностей на основе бизнес-функций;
- преобразование матрицы приложений, содержащих сущности данных, к виду, позволяющему установить последовательность реализации, определяемую данными с помощью соответствующей оптимизационной процедуры;

- формирование критериев (количественных и качественных) последовательности реализации;
- формирование последовательности модификации существующих систем и приобретения технологий.

Остальные шаги этапа традиционны для задачи планирования.

На этапе **«Заключительное планирование»** осуществляется подготовка окончательного отчета по архитектуре предприятия, подготовка и проведение презентации.

Этап **«Переход к реализации»** включает следующие шаги:

- 1) планирование перехода (спецификацию целей перехода, формирование плана перехода, назначение ответственности за переход, определение руководителя-лидера);
- 2) адаптацию подхода (методологии, инструменты);
- 3) инвентаризацию компьютерных ресурсов (приобретение необходимого оборудования, обеспечение надежности хранилища);
- 4) чистку архитектуры (ревизию, добавление деталей и обновление);
- 5) изменение организационно-штатной структуры;
- 6) набор персонала;
- 7) проведение обучения;
- 8) введение стандартов на программирование;
- 9) введение процедурных стандартов;
- 10) разработку детальных планов по приложениям;
- 11) определение и утверждение даты завершения перехода.

Эти шаги являются традиционными и не рассматриваются в рамках данного учебника.

Данный подход помог многим компаниям и государственным ведомствам в организации процесса моделирования, стратегического бизнес-планирования, реорганизации деловых процессов, проектирования различных систем, выработки стандартов на данные, управления проектами. В частности, этой методикой пользовались такие организации, как Federal Express, Министерство энергетики США, Штаб Военно-воздушных сил США. Например, в Министерстве энергетики США основная фаза процесса разработки архитектуры заняла примерно 6 месяцев [1].

Если «наложить» метод ЕАР Спивака на модель Захмана, то можно сказать, что метод ЕАР является руководством по заполнению первых двух строк таблицы Захмана, которые описывают контекст архитектуры и концептуальную модель бизнеса предприятия. Проектирование систем, которое начинается с третьей строки таблицы Захмана, остается за рамками метода Спивака [1].

4.10. Краткое сравнение различных методик

Несмотря на формальное наличие стандартов в области описания архитектуры (ISO, IEEE, The Open Group и т. д.), ни одна из известных методик не имеет доминирующего положения в плане своего использования. Например, опрос, проведенный в 2003 г. Институтом разработки корпоративной архитектуры (Institute for Enterprise Architecture Developments), показал, что собственные методики использовали около 32 % организаций, модель Захмана — 20 %, остальные методики — не более 5–6 % респондентов [1]. Таким образом, основная рекомендация состоит в использовании всего лучшего, что предлагается различными методиками, с учетом их достоинств и недостатков. При этом процесс разработки архитектуры предприятия необходимо начинать с четкого осознания целей по организации бизнеса.

Из множества существующих методик и моделей разработки архитектуры предприятия **модель Захмана** является наиболее используемой. Она имеет безусловную ценность для архитекторов, хотя и с явными ограничениями. Для многих клеток матрицы Захмана корпоративные архитекторы определяют, в лучшем случае, шаблоны проектирования, а не продукты описания архитектуры в полном смысле этого слова. Например, в клетке на пересечении столбца «ЧТО» и строки «Физический уровень» определение архитектуры заканчивается стандартным сервером приложений. Есть также некоторые относящиеся к описанию архитектуры документы, которые непонятно куда помещать с

точки зрения классификации матрицы Захмана (например, общая стратегия развития прикладных систем на ближайшие годы). С другой стороны, верхние уровни модели Захмана обеспечивают весьма полезную структуру для совместного обсуждения проблем архитектуры предприятия с бизнес-руководством [1].

Архитектурные **методики Gartner** отличаются глубиной концептуального взгляда на проблему, способностью данной концепции использовать практический опыт большого количества клиентов, наличием элементов, позволяющих определять основные направления развития технологий в различных предметных областях, связанных с информационными технологиями. Однако в методике отсутствуют, по крайней мере в публичном доступе, детальные описания, примеры и руководства, упрощающие практическое использование представлений Gartner об архитектуре.

Детальные описания **методики META Group** содержат достаточно подробные описания различных представлений архитектуры и документов, включая их шаблоны, которые должны создаваться в процессе работы над созданием архитектуры. В методике акцентируется внимание на организации архитектурного процесса и его связи с другими аспектами управления ИТ, в частности с управлением корпоративными проектами. Но эти материалы, так же, как и Gartner, отсутствуют в публичном доступе.

Основной областью внимания **методики TOGAF** является архитектурная методология, существенно расширенная в своей последней версии за рамки технологической архитектуры. Теперь в методике рассматривается бизнес-архитектура, архитектура данных и архитектура приложений. В настоящее время это одна из наиболее полных методик, доступная бесплатно.

Методика NASCIO содержит описания и образцы процессов, используемых для управления архитектурой и ее жизненным циклом, а также форматы и примеры документов с описанием технологической архитектуры. Отличительной особенностью данной методики является наличие архитектуры решений.

Использование сценариев в **модели «4+1»** позволяет объединить четыре представления архитектуры предприятия и выделить наиболее важные требования, которым должна удовлетво-

рять система. Такое объединяющее представление может быть избыточным, но оно позволяет проверить работоспособность и полноту разрабатываемой архитектуры с помощью предоставления наглядной иллюстрации всего процесса построения.

Методика разработки SAM предлагает интересный инструмент анализа деятельности предприятия, категоризации и связывания между собой различных элементов описания архитектуры. Методика также содержит интересные находки в плане идентификации изменяющихся и относительно стабильных областей архитектуры [1].

Сильными сторонами архитектурных **методик Microsoft** является их практическая близость к предметной области разработки архитектуры и эксплуатации сложных программных систем. В данных методиках реализованы командные способы организации работы при построении архитектуры предприятия. Описание представлений об архитектуре в соответствии с методиками Microsoft в какой-то мере основано на модели Захмана. Документы находятся в публичном доступе, что усиливает привлекательность данных методик для разработчиков.

Метод планирования архитектуры организации EAP также является одним из наиболее полных методов, позволяющих на качественном уровне представить архитектуру предприятия, основываясь на его функциональной модели. Данный метод охватывает все шаги проектирования от момента инициации планирования архитектуры до начала процесса ее реализации.

В табл. 4.6 приведены сравнительные характеристики вышечисленных методов и моделей построения архитектуры предприятия. На практике находят применение и другие средства представления архитектуры предприятия.

Методика Федеральной архитектуры правительства США FEAF, разработка которой началась в конце 1990-х годов, содержит хороший обзор видения и принципов архитектуры предприятия. Документы, содержащие общее описание методике FEAF, имеют ссылки на некоторые другие методики, например модель Захмана.

Таблица 4.6
Сопоставление возможностей различных моделей и методик построения архитектуры предприятия

Предоставляемая возможность	Наличие возможности в концепции построения архитектуры предприятия									
	Модель Захмана	Модель Gartner	Методика META Group	Методика TOGAF	NASCIO Toolkit	Модель «4+1»	Модель SAM	Методики Microsoft	Методики EAP	Метод EAP
Иерархический подход, возможность связи с бизнес-стратегией	+	+	+	—	—	+	+	+	+	+
Поддержка различных уровней абстракции	+	+	+	+	+	+	+	+	+	+
Формальный язык и система обозначений	—	—	—	+	—	—	—	—	—	—
Описание процесса разработки архитектуры	—	+	+	+	—	+	—	+	+	+
Рекомендации по управлению архитектурой	—	+	+	+	+	—	—	+	+	+

Методика FEAF содержит четыре представления (бизнес, информация, приложения, инфраструктура) и пять справочных моделей для их описания. Сильной ее стороной является детальная проработка каждого из представлений. Интересны и аспекты, связанные с показателями эффективности (часть бизнес-архитектуры) и отслеживанием связей между этими показателями и использованием информационных технологий [1].

TEAF (Treasury Enterprise Architecture Framework) — архитектура Казначейства США, построенная на основе методики федеральной архитектуры государственных организаций FEAF, но с более глубоким и детальным уровнем проработки многих вопросов, поскольку методика разрабатывалась для отдельной организации. В состав TEAF включены шаблоны документов для различных направлений деятельности. Методика TEAF содержит много хороших примеров архитектурных принципов и документов, создаваемых в результате работы над архитектурой. Методика предлагает для категоризации документов и моделей описания архитектуры упрощенную матрицу размером «4 x 4» вместо «6 x 6» в модели Захмана и содержит указания о принадлежности разрабатываемых моделей к определенным ячейкам таблицы.

C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance)¹ — методика построения архитектуры, разработанная в Министерстве обороны США в 1996–98 гг. С декабря 2003 г. ее заменила рамочная архитектура **DoDAF** (Department of Defence Architecture Framework), аналогичная TEAF по уровню детализации и наличию большого количества примеров моделей и документов, используемых для описания архитектуры. Однако она идет дальше TEAF в том плане, что приводит большое количество фактических примеров этих моделей и документов (а не только шаблоны).

DoDAF содержит правила, руководства и продукты (документы или артефакты), которые должны использоваться при разработке и описании архитектуры различных систем, используемых

¹ Переводится на русский язык как «Командование, Управление, Коммуникации, Компьютеры, Информация, Наблюдение и Разведка»

военными ведомствами США. Все это, по мнению разработчиков, улучшает возможности информационных технологий с точки зрения быстрой мобилизации для выполнения военных операций. Особенностью методики является предоставляемая возможность сравнения, анализа и интеграции архитектур систем, используемых как в различных функциональных подразделениях, так и в географически распределенной организационной среде.

Наличие в методике большого количества примеров и высокий уровень детализации всех описаний делают ее отличным учебным пособием, в котором раскрывается понятие архитектуры предприятия и приводится характеристика содержащихся в архитектуре документов и описаний.

В соответствии с методикой DoDAF описание архитектуры содержит три различных представления: *операционное, системное и представление технических стандартов*. Каждое из представлений используется для отражения различных архитектурных характеристик и атрибутов, хотя между ними имеются определенные пересечения. Некоторые из атрибутов объединяют два различных представления, что обеспечивает целостность, единство и единообразие в описании архитектуры. Наиболее полезным является *интегрированное* описание архитектуры, сочетающее различные представления в описании систем.

Достаточно интересной моделью является **RM-ODP** (Reference Model of Open Distributed Processing — Справочная Модель Открытых Распределенных Вычислений)¹, принятая Международной организацией стандартизации ISO в качестве стандарта в четырех частях X.901, X.902, X.903 и X.904. Модель RM-ODP, на которую также ссылаются как на ISO/IEC 10746, имеет много общего с IEEE 1471 в плане определения метамодели архитектуры, но идет гораздо дальше в отношении определений и документирования специфических принципов. В основе этой модели лежат принципы анализа систем в разрезе нескольких представлений и объектно-ориентированная парадигма создания

¹ Данная модель используется для описания архитектуры электронного правительства Германии.

систем. Модель RM-ODP — одна из наиболее полных с точки зрения набора различных представлений, используемых для описания архитектуры системы. Важными для модели RM-ODP понятиями являются «представления», «функции» и «средства обеспечения прозрачности распространения» (distribution transparencies).

Модель определяет **пять представлений (viewpoints)**:

1) *корпоративное представление*, описывающее цели, масштабы (границы), процессы и политики, связанные с созданием прикладных систем;

2) *информационное представление*, отражающее характеристики и семантику обрабатываемых данных, т. е. модель данных;

3) *вычислительное представление*, демонстрирующее декомпозицию прикладной системы на функциональные модули и интерфейсы взаимодействия;

4) *проектировочное представление*, определяющее распределение отдельных элементов системы по физическим ресурсам и связи между ними;

5) *технологическое представление*, описывающее технологии, используемые для создания прикладных систем.

Кроме представлений RM-ODP содержит так называемые функции. Всего выделено четыре функции:

1) *функция управления*, определяющая способы управления системой, начиная с уровня узлов (серверов) и вплоть до объектов, выполняемых на этих узлах;

2) *функция координации*, детализирующая вопросы взаимосвязи событий в системе;

3) *функция репозитория*, описывающая организацию хранения информации;

4) *функция безопасности*, описывающая вопросы управления безопасностью в системе, а также методы авторизации доступа, обеспечения целостности, аудита, управления правами доступа.

RM-ODP выделяет восемь так называемых средств обеспечения прозрачности: *прозрачность доступа, сбоев, местоположения, миграции, сохранения, перераспределения, репликации и транзакций*.

Компанией Philips разработана модель **SAFCR** (Customer Objectives — Задачи Заказчика, Application — Приложения, Functional — Функциональность, Conceptual — Концепция и Realization — Реализация). Она первоначально предназначалась для разработки архитектур встроенных (embedded) систем, но позднее стала применяться и для более сложных комплексов — вплоть до глобальных систем управления автомобильным движением.

Аббревиатура SAFCR образована из названий пяти основных представлений модели. Первые два представления (Customer Objectives и Application) помогают ответить на вопрос «зачем?», т. е. описывают цели создания системы. Функциональное представление (Functional) отвечает на вопрос «что должна система выполнять?». Несмотря на название, в данное представление входят также и нефункциональные требования, такие как, например, требуемый интерфейс пользователя или выбор используемой СУБД. Последние два представления (Conceptual и Realization) отвечают на вопрос «как должно осуществляться функционирование системы?». При этом концептуальное представление со временем подвергается незначительным изменениям по сравнению с представлением, описывающим реализацию. Принцип организации этих представлений в целом соответствует выбору столбцов в модели Захмана.

Основной задачей архитектора в этом случае является создание образа системы, назначение которого состоит в обеспечении согласованной и сбалансированной интеграции этих представлений с целью построения системы, отвечающей следующим основным условиям:

- предоставление возможностей, имеющих определенную ценность для заказчика с точки зрения выполнения предъявляемых требований;
- реализуемость в конкретной среде заказчика;
- гарантированность определенного уровня оптимальности по эффективности, стоимости и удобству использования.

Другой проект компании Philips, названный «Гауди» в честь выдающегося испанского архитектора (это еще раз подчеркивает

аналогию между традиционной архитектурой и архитектурой информационных систем), имеет еще более амбициозные цели — сформулировать рекомендации по созданию хорошей архитектуры и разработать систему подготовки архитекторов, которые смогут создавать такие системы. В данном случае определяющими являются именно понятия критериев «хорошей архитектуры», которые должны включать как объективные показатели, так и субъективное восприятие системы или продукта [1].

Вопросы для самоконтроля

1. Назовите модели и методики, основой для которых послужила модель Захмана.
2. Какие цели реализуются в модели Захмана?
3. Дайте описание таблицы (матрицы) модели Захмана.
4. Какие правила предъявляются к заполнению таблицы (матрицы) Захмана?
5. Сформулируйте уровни модели Gartner 2002.
6. Опишите этапы методики META Group.
7. Опишите фазы ADM методики TOGAF.
8. Опишите структурную схему методики NASCIO Architecture Toolkit.
9. Опишите представления модели «4+1».
10. Опишите типичные сферы интересов модели SAM.
11. Опишите методики Microsoft, которые используются для построения архитектуры предприятия.
12. Опишите этапы планирования архитектуры предприятия в соответствии с методом EAP.
13. Сопоставьте возможности различных моделей построения архитектуры.

Заключение

Представленный в пособии анализ существующих средств построения архитектуры предприятия позволяет сделать вывод об отсутствии на сегодняшний день единого стандарта на описание архитектуры. Разработчики могут использовать различные методологии, модели и методики для создания собственной концепции представления архитектуры предприятия, которая будет уникальной для описываемой организации. При этом в процессе создания архитектуры общие концепции и структуры описания архитектуры используются в качестве базовых при реализации имеющихся моделей и методик построения архитектуры, что существенно сокращает время обучения специалистов и создает общую платформу в создании единой стратегии разработки. Кроме того, результат адаптации методики под нужды конкретного предприятия отражает характерные именно для данного предприятия моменты, связанные с организацией работы управленческого персонала, уровнем финансирования, наличием ресурсов и т. п.

Важным моментом при создании методики, ориентированной на потребности отдельного предприятия, является необходимость включения, по возможности, максимального описания всех используемых сокращений и графических интерпретаций, но их количество не должно превышать возможности реализации.

Создание новых методик построения архитектуры предприятия находится в процессе постоянного развития и совершенствования. Особенно это характерно для зарубежного рынка, хотя и на отечественном рынке по мере осознания необходимости и полезности внедрения стандартов серии ISO 9000¹, явившегося первым шагом к построению архитектуры предприятия, руководители многих организаций задумались о неизбежности построе-

¹ Система стандартов менеджмента качества разработана Техническим комитетом Международной Организации по Стандартизации (International Organization for Standardization).

ния архитектуры предприятия, создающей возможности для повышения конкурентоспособности организации.

Для поддержания процесса развития знаний в данной предметной области на современном уровне (период устаревания составляет 2–3 года) необходимо регулярно изучать информацию, представленную в электронных ресурсах организаций, распространяющих методики построения архитектуры, а также просматривать следующие интернет-ресурсы¹:

- on-line библиотека (<http://www.citforum.ru/>);
- электронное издание журнала «Открытые системы» (<http://www.osp.ru/>);
- электронное издание журнала «Intelligent Enterprise/Корпоративные системы» (<http://www.iemag.ru/>);
- сайт компании IBM — лидера в области разработки аппаратного и программного обеспечения (<http://www.ibm.com/>);
- сайт компании Microsoft — лидера в области разработки программного обеспечения (<http://www.microsoft.com/>);
- сайт компании BearingPoint — консультанта в области управления и ИТ (<http://www.bearingpoint.ru/>).

¹ Это далеко неполный перечень ресурсов, посвященных в той или иной мере архитектуре предприятия.

Литература

1. Данилин А.В., Слюсаренко А.И. Архитектура предприятия [Электронный ресурс]: курс лекций // Интернет-университет информационных технологий: [сайт]. [2003–2010]. — URL: <http://www.intuit.ru/department/itmngt/entarc/0/>
2. Современный экономический словарь / Б.А. Райзберг [и др.]. — 5-е изд., перераб. и доп. — М.: ИНФРА-М, 2007. — 495 с.
3. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++: пер. с англ. — 2-е изд. — М.: Бином; СПб.: Невский диалект, 1998. — 560 с.
4. Васильев Р.Б., Калянов Г.Н., Левочкина Г.А. Управление развитием информационных систем: учеб. пособие / под ред. Г.Н. Калянова. — М.: Горячая линия-Телеком, 2009. — 376 с.
5. Свободная энциклопедия Википедия. — URL: <http://ru.wikipedia.org/> (дата обращения: 12.03.2011).
6. ISO 15704. Industrial automation systems — Requirements for enterprise-reference architectures and methodologies // International Organization for Standardization: [сайт]. — URL: http://www.iso.org/so/iso_catalogue/catalogue_tc/catalogue_detail.htm?cs
7. Federal Enterprise Architecture Framework, Version 1.1, September // The Chief Information Officers Council (USA): [сайт]. — URL: http://www.cio.gov/documents_details.cfm/uid/1F432311-2170-9AD7-F2053C10765E0E1C/structure/Enterprise%20Architerture/category/
8. Компания Gartner [Электронный ресурс]: [сайт]. — URL: <http://www.gartner.com> (дата обращения: 16.04.2011)
9. IEEE 1471-2000-IEEE Recommended Practice for Architectural Description for Software-Intensive Systems // Institute of Electrical and Electronics Engineers: [сайт]. — URL: <http://standards.ieee.org/findstds/standard/1471-2000.html>
10. Галактионов В. Системная архитектура и ее место в архитектуре предприятия [Электронный ресурс] // Директор ИС. — 2002. — № 1(05.02.2002). — URL: <http://www.osp.ru/cio/2002/05/>
11. Виханский О.С. Стратегическое управление: учебник. — 2-е изд., перераб. и доп. — М.: Гардарика, 1998. — 296 с.

12. Економіка підприємства: Підручник / за заг. ред. д-ра екон. наук, проф. С. Ф. Покропивного. — Киев: КНЕУ, 2003. — 608 с.
13. Быкова А. Организационные структуры управления. — М.: ОЛМА-ПРЕСС Инвест, 2003. — 160 с.
14. Carbone J. IT Architecture Toolkit. — Prentice Hall PTR, 2004. — 256 p.
15. TRM FEAF (Federal Enterprise Architecture Framework): техническая справочная модель методики Федеральной архитектуры США [Электронный ресурс]: [сайт]. — URL: <http://www.whitehouse.gov/omb/egov/>.
16. Калянов Г.Н. Консалтинг при автоматизации предприятий (подходы, методы, средства). — М.: СИНТЕГ, 1997. — 316 с.
17. Ковалев С.М., Ковалев В.М. Современные методологии описания бизнес-процессов — просто о сложном [Электронный ресурс] // Консультант директора. — 2004. — № 12. — URL: <http://conti.kuzbass.net/education/articles/kovalev5/>.
18. Верников Г. Описание стандарта документирования технологических процессов IDEF3 [Электронный ресурс] // [Персональная страница Г. Верникова]. — URL: <http://idefinfo.ru/content/view/18/51/> (дата обращения: 16.04.2011).
19. Верников Г. Основы методологии IDEF1X [Электронный ресурс]. — URL: <http://www.interface.ru/fset.asp?url=/ca/idef1x.htm> (дата обращения: 21.03.2011).
20. Seidewitz E., Stark M. Towards a General Object-oriented Software Development Methodology // ACM Digital Library: [сайт]. — URL: <http://portal.acm.org/citation.cfm?id=25315>
21. Robson D. Object-oriented Software Systems / Byte. — Vol. 6(8). — P. 74.
22. Леоненков А.В. Нотация и семантика языка UML [Электронный ресурс]. — URL: <http://www.intuit.ru/department/pl/umlbasics/1/> (дата обращения: 21.03.2011).
23. Model Maker 9: инструментальное средство IDEF1X [Электронный ресурс] // Компания Model-Maker Tools BV: [сайт]. — URL: www.modelmakertools.com.
24. Together 2008: CASE-средство [Электронный ресурс] // Компания Borland: [сайт]. — URL: www.borland.com.

25. Инструментарий ARIS. Методы. Версия 4.1 [Электронный ресурс] // Весть — МетаТехнология. — 2000. — Апрель. — 227 с. — URL: <http://vernikov.ru/biznes-modelirovanie/tehnologii-i-standarty/item/245-aris41methods.html> (дата обращения: 03.12.2010).

26. Репин В. Сравнительный анализ нотаций ARIS eEPC / IDEF0, IDEF3 1 [Электронный ресурс]. [Сайт]. — [2009]. — URL: <http://vernikov.ru/biznes-modelirovanie/tehnologii-i-standarty/item/220-sravnitelnyi-analiz-notacii-aris-eepec-idef0-idef3.html> (дата обращения 09.10.2010).

27. Зиндер Е.З. «3D-предприятие» — модель стратегии трансформирующейся системы [Электронный ресурс]. [Сайт]. — URL: http://www.sept2000.ru/page_attachments/0000/0003/3d-p.doc (дата обращения: 13.09.2010).

28. Extended Enterprise Architecture Framework (E2AF) / Information Exchange Area of the Institute For Enterprise Architecture Developments. — URL: <http://www.enterprise-architecture.info>

29. Four Domains architecture / The ACM Digital Library is published by the Association for Computing Machinery. [Site]. [2011]. — URL: <http://www.stratetect.com/2008/08/the-four-domain-architecture-discovered>.

30. TOGAF Version 9 «Enterprise Edition» // The Open Group. — URL: <http://www.opengroup.org/togaf>

31. NASCIO. Enterprise Architecture Development Tool-Kit v.3.0 // NASCIO. — URL: <http://www.nascio.org/resources/faresources.cfm>.

32. Kruchten Ph. Architectural Blueprints «The 4+1 View Model of Software Architecture» // IEEE Software. — November 1995. — N. 12 (6). — P. 42-50.

33. Teale Ph., Jarvis R. Business Patterns for Software Engineering Use. P. 1 // MSDN: [Site]. [2004]. — URL: <http://msdn.microsoft.com/ru-ru/library/aa480022.aspx>.

34. Методики фирмы Microsoft // Microsoft Architecture Journal: электронный журнал. — URL: <http://msdn.microsoft.com/architecture/journ/>.

35. Steven H. Spewak. Enterprise Architecture Planning. — N.Y.: John Wiley&Sons Inc., 2003. — 314 p.

Приложение

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Задание 1

Формирование миссии и стратегии предприятия

Выполнение задания состоит из трех этапов.

I. Формирование общего представления о предприятии

1. Изучите совокупность факторов, влияющих на организационную структуру исследуемого предприятия¹.
2. Охарактеризуйте продукцию, выпускаемую предприятием.
3. Сформулируйте существующую стратегию предприятия. По-пробуйте ответить на вопрос: какой цели подчинялась работа предприятия в последние 4–5 лет.

II. Определение внутренних и внешних факторов, влияющих на развитие предприятия

1. Определите мотивацию потребителей продукции выбранного предприятия.
2. Определите доминирующую мотивационную характеристику персонала рассматриваемого предприятия, поскольку мотивация работника во многом определяет результативность его труда. Попробуйте ответить на вопрос: для чего люди работают на данном предприятии.
3. Определите доминирующий мотив внешних инвесторов предприятия.

¹ В качестве исследуемого предприятия может рассматриваться вариант, предложенный преподавателем (см. список предлагаемых предприятий в конце задания), либо выбранный студентом самостоятельно.

4. Оцените макроэкономические тенденции, оказывающие влияние на деятельность предприятия. В какой степени благополучие (неблагополучие) предприятия определяется общеэкономической конъюнктурой и политикой правительства?

5. Определите положение анализируемого предприятия на рынках сбыта и ресурсов. Является ли данное предприятие монополистом на своем рынке? Если нет, то ощущает ли оно конкуренцию и с чьей стороны? Какие факторы ограничивают сбыт профильной продукции?

III. Характеристика процессов предприятия

1. Сформулируйте основные принципы политики рассматриваемого предприятия по отношению к спросу (потребителям). Меняется ли что-то в политике производства, сбыта и цен, если происходят изменения на рынке сбыта? Какие могут быть приняты шаги в ответ на рост или падение спроса?

2. Сформулируйте основные принципы кадровой политики предприятия. Меняется ли что-то в кадровой политике предприятия, если происходят изменения на рынке и в производстве? Важна ли для вас производительность вашего предприятия? Если на нем имеется избыточная рабочая сила, то чем это вызвано?

3. Оцените направления структурных изменений, произошедших на вашем предприятии за последние три–пять лет. Как изменение сбытовой политики в результате действия рыночных факторов сказалось на структуре производимой предприятием продукции? Изменилось ли и как использование материальных ресурсов после падения загрузки производства?

4. Охарактеризуйте применяемую на вашем предприятии технологию. Получает ли предприятие экономию на масштабах производства (т. е. растет ли прибыль на единицу продукции при увеличении объемов производства и сбыта)? Позволяет ли технология перейти с выпуска одного вида продукции на выпуск другого, с какими затратами и в какие сроки?

5. Охарактеризуйте стиль управления на вашем предприятии. Доверяете ли вы своим подчиненным или предпочитаете быть постоянно в курсе их дел, чтобы застраховаться от их ошибок (непрофессионализма)?

6. Оцените состояние капитала и инвестиционную привлекательность предприятия. Выберите и классифицируйте основные факторы, отрицательно влияющие на состояние и структуру капитала. Можно ли нейтрализовать эти факторы и каким образом?

7. Оцените перспективы дальнейшего существования предприятия в рамках сложившейся стратегии. Достаточно ли жизнеспособна, с вашей точки зрения, организационная структура, которая использовалась до сих пор на вашем предприятии? Если нет, то, в каком направлении она может быть усовершенствована? Каковы основные проблемы, возникающие в связи с этим?

Результаты выполнения данного и нижеприведенных заданий оформляются в виде отчета.

Список предлагаемых к рассмотрению предприятий:

- 1) супермаркет;
- 2) крупное сельскохозяйственное предприятие;
- 3) нефтехимическое предприятие;
- 4) металлургический завод;
- 5) швейная фабрика;
- 6) инновационное предприятие по выпуску научно-технической продукции;
- 7) автомобильный концерн;
- 8) высшее учебное заведение;
- 9) концертный зал, театр или кинотеатр;
- 10) кондитерская фабрика;
- 11) электростанция;
- 12) транспортная компания;
- 13) банк;
- 14) туроператор;
- 15) страховая фирма;
- 16) строительное предприятие;
- 17) биржа;

- 18) крупное предприятие оптовой торговли с выходом на международный рынок;
- 19) предприятие по добыче полезных ископаемых;
- 20) предприятие по производству военного вооружения.

Задание 2

Выявление технико-экономических факторов формирования организационной структуры предприятия

1. Создайте схему организационной структуры выбранного предприятия. При этом выделите следующие элементы:

- *технологическую структуру* (подразделения основного производства и взаимосвязи между ними);
- *производственную структуру* (подразделения вспомогательного производства и взаимосвязи между ними);
- *хозяйственную структуру* (подсобные хозяйства, производства из отходов и объекты социальной сферы);
- *организационную структуру* (состав и взаимосвязи подразделений управления предприятием).

2. Сформулируйте критерии эффективности работы подразделений.

3. Проанализируйте наиболее значимые факторы, влияющие на текущие результаты (прибыль) работы подразделений вашего предприятия. Позволяют ли текущие результаты хозяйственной деятельности подразделения возместить сделанные затраты в сложившейся хозяйственной структуре? Если нет — каковы причины неэффективности? Связана ли она с принципиальной неконкурентоспособностью продукции (услуг) подразделения или является результатом нерациональных связей с другими подразделениями?

Задание 3

Построение функциональной модели предприятия с использованием методологий структурного анализа и проектирования. Модель AS-IS

1. Разработайте функциональную модель выбранного предприятия с использованием методологий структурного анализа и проектирования IDEF. Постройте модель **AS-IS (как есть)**.
2. Оцените полезность использования методологий IDEF0, DFD, IDEF3 при построении модели вашего предприятия.
3. Приведите обоснование точки зрения, выбранной вами при построении модели предприятия.
4. Выберите наиболее важный или интересный бизнес-процесс в деятельности вашего предприятия, нуждающийся в изменении. Обоснуйте свой выбор.

Задание 4

Построение функциональной модели предприятия с использованием методологий структурного анализа и проектирования. Модель TO-BE

1. Разработайте модель наиболее важного или интересного бизнес-процесса в деятельности вашего предприятия, выбранного в задании 3, с использованием методологий структурного анализа и проектирования IDEF. Постройте модель **TO-BE (как должно быть)**.
2. Оцените полезность использования методологий IDEF0, DFD, IDEF3 при построении модели данного процесса.
3. Приведите обоснование точки зрения, выбранной вами при построении модели процесса. Докажите необходимость предлагаемого вами изменения.

Задание 5

Моделирование предприятия с использованием методологий объектно-ориентированного анализа и проектирования

1. Постройте модель работы выбранного предприятия с использованием методологий UML.
2. Оцените полезность использования диаграмм методологии UML при построении модели вашего предприятия.
3. Какие типы диаграмм, на ваш взгляд, являются наиболее значимыми при построении модели предприятия.

Задание 6

Построение системной архитектуры предприятия. Архитектура информации

1. Определите тип и объем необходимой информации, которая должна быть предоставлена для осуществления процессов, описанных в предыдущих практических заданиях, ответственными за их выполнение сотрудниками.
2. Покажите связь между понятиями «архитектура информации» и «архитектура данных».
3. Постройте модели информации вашего предприятия на различных уровнях абстракции.

Задание 7**Построение
системной архитектуры предприятия.
Архитектура приложений**

1. Опишите имеющийся на выбранном предприятии портфель прикладных систем.
2. Представьте планируемый портфель прикладных систем данного предприятия.
3. Составьте план миграции прикладных систем.
4. Приведите обоснование используемой вами модели для построения архитектуры приложений вашего предприятия.

Задание 8**Построение
системной архитектуры предприятия.
Технологическая архитектура**

1. Представьте технологическую архитектуру вашего предприятия в разрезе следующих технологий:
 - аппаратные платформы;
 - операционные системы;
 - системы управления базами данных;
 - средства разработки;
 - языки программирования;
 - сервисы электронной почты;
 - системы безопасности;
 - сетевая инфраструктура и т. д.
2. Укажите технологии, являющиеся наиболее важными, на ваш взгляд.

Задание 9

Формирование многослойной архитектуры предприятия

1. Представьте комплексную модель архитектуры выбранного предприятия в разрезе следующих слоев архитектуры:

- фронт-офис (Front-Office);
- мидл-офис (Middle-office);
- бэк-офис (Back-office);
- учёт (Accounting);
- информационное хранилище (Data Warehouse);
- отчетность (Reporting).

2. Определите, все ли слои архитектуры предприятия присутствуют в построенной вами архитектуре и какие из них являются наиболее важными для бизнеса.

Оглавление

Предисловие	3
Введение	4
ГЛАВА 1. АРХИТЕКТУРА ПРЕДПРИЯТИЯ В РАЗЛИЧНЫХ АСПЕКТАХ	
1.1. Сущность и базовые понятия дисциплины	8
1.1.1. Предприятие как объект изучения	8
1.1.2. Понятие архитектуры предприятия	16
1.1.3. Значение архитектуры предприятия в современных условиях	20
1.2. Статический и динамический аспекты архитектуры предприятия	22
1.2.1. Основные элементы и слои архитектуры предприятия	22
1.2.2. Миссия и стратегическое планирование	30
1.2.3. Бизнес-архитектура	39
1.2.4. Системная структура предприятия	55
ГЛАВА 2. КЛАССИЧЕСКИЕ МЕТОДОЛОГИИ ПОСТРОЕНИЯ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ	
2.1. Общие принципы построения архитектуры предприятия	71
2.2. Методологии структурного анализа и проектирования	74
2.2.1. Структурный анализ	74
2.2.2. Методология на основе диаграмм потоков данных DFD	77
2.2.3. Методология структурного анализа и проектирования IDEF0	82
2.2.4. Методология моделирования и стандарт документирования процессов IDEF3	88
2.2.5. Методология моделирования отношений между данными IDEF1X	94

2.3. Методология объектно-ориентированного анализа и проектирования	102
2.3.1. Объектная модель	102
2.3.2. Язык моделирования UML	106
2.3.3. Паттерны	113
ГЛАВА 3. МЕТОДОЛОГИЯ ARIS ДЛЯ ПОСТРОЕНИЯ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ	
3.1. Основы методологии ARIS	124
3.2. Организационная модель ARIS	129
3.3. Функциональная модель ARIS	133
3.4. Информационная модель ARIS	138
3.5. Управляющая модель ARIS	146
3.6. Модели ресурсов ARIS	158
3.7. Метод управления знаниями в методологии ARIS	165
3.8. Сравнительный анализ методологий ARIS и IDEF	172
ГЛАВА 4. ОБЗОР МОДЕЛЕЙ И МЕТОДИК ПОСТРОЕНИЯ АРХИТЕКТУРЫ ПРЕДПРИЯТИЙ	
4.1. Модель Захмана	185
4.2. Модель описания ИТ-архитектуры Gartner	193
4.3. Методика META Group	196
4.4. Методика TOGAF	201
4.5. NASCIO Architecture Toolkit	204
4.6. Модель представления архитектуры «4+1»	211
4.7. Стратегическая модель архитектуры SAM	214
4.8. Архитектурные концепции и методики Microsoft	217
4.9. Метод планирования архитектуры организации EAP	223
4.10. Краткое сравнение различных методик	235
Заключение	244
Литература	246
Приложение. Индивидуальные задания	249
Задание 1. Формирование миссии и стратегии предприятия	249
Задание 2. Выявление технико-экономических факторов формирования организационной структуры предприятия	252

Задание 3. Построение функциональной модели предприятия с использованием методологий структурного анализа и проектирования. Модель AS-IS	253
Задание 4. Построение функциональной модели предприятия с использованием методологий структурного анализа и проектирования. Модель TO-BE	253
Задание 5. Моделирование предприятия с использованием методологий объектно-ориентированного анализа и проектирования	254
Задание 6. Построение системной архитектуры предприятия. Архитектура информации	254
Задание 7. Построение системной архитектуры предприятия. Архитектура приложений	255
Задание 8. Построение системной архитектуры предприятия. Технологическая архитектура	255
Задание 9. Формирование многослойной архитектуры предприятия	256

УЧЕБНОЕ ИЗДАНИЕ
Гриценко Юрий Борисович
АРХИТЕКТУРА ПРЕДПРИЯТИЯ

Учебное пособие

Редактор Н.В. Коновалова
Корректор О.В. Полещук

Подписано в печать 30.06.14. Формат 60x84/16.

Усл. печ. л. 15,11. Тираж 200 экз. Заказ 577.

Томский государственный университет систем управления
и радиоэлектроники,
634050, г. Томск, пр. Ленина, 40. Тел. (3822) 53-30-18.